OpenOffice.org » GP2X » Сети

LINUX FORMAT

Главное в мире Linux

Август 2009 № 8 (121)

LXFDVD

Linux Mint 7

Ubuntu на новом уровне: полная версия

ПЛЮС: RFRemix 11 » Ulteo 1.0 » EduMandriva

Выбираем лучший нетбук

8 Недорогие, небольшие, с Linux — найдите тот, что подойдет вам

38 страниц учебников для всех!

Git для /etc

Машина времени для системных настроек с. 28

Интернет из консоли

Бороздите просторы Сети, не запуская X-сервер с. 30

Вещайте музыку

Медиа-сервер для вашей домашней сети с. 46

Поговорим?

Хитроумный Jabber-бот на Python с. 66

Защита серверов

Nessus, Tripwire и Iptables обезопасят от угроз с. 80

«Наверное, так же почувствовал себя Левенгук, впервые заглянув в микроскоп.»

Крис Браун недаром имеет ученую степень с. 42

В каталоге агентства «Роспечать» — подписной индекс 20882
В каталоге «Пресса России» — подписной индекс 87974





Что мы делаем

» Мы поддерживаем открытое сообщество, предоставляя источник информации и площадку для обмена мнениями.

» Мы помогаем всем читателям получить от Linux максимум пользы, публикуя статьи в разделе «Учебники»: здесь каждый найдет что-то по своему вкусу.

» Мы выпускам весь код, появляющийся на страницах раздела «Учебники», по лицензии GNU GPLv3.

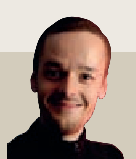
» Мы стремимся предоставлять точные, актуальные и непредвзятые сведения обо всем, что касается Linux и свободного ПО.

Кто мы

После ярмарки нетбуков на странице 16, у нас остался только один вопрос к команде LXF: если бы вы могли добавить единственную функцию к нетбуку всех времен и народов, то что бы это было?



Грэм Моррисон
Я бы заменил web-камеру устройством, определяющим невольное расширение радужки глаза.



Майк Сондерс
Этаноловые батареи с пивными элементами питания. Одну – тебе, одну – мне...



Нейл Ботвик
Сберег бы ресурсы, позволяя запускать не больше 3-х программ сразу. Можно запатентовать идею?



Эфраин Эрнандес-Мендоса
Фазированная плазменная 40-ваттная винтовка. Дробовик с лазерным прицелом. 9-мм «Узи».



Эндрю Грегори
Апплет, измеряющий, сколько вы выпили, и вычисляющий, много ли теперь потребуется закуски.



Энди Ченнел
Я бы ничего не добавлял, а просто взял имечко позвучнее, вроде «дешевый маленький ноутбук».



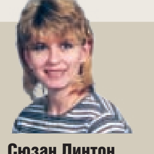
Дэвид Картрайт
Распознавание отпечатков пальцев. Не получится – тогда лазер с акулой на голове.



Энди Хадсон
Радиомаяк, чтобы можно было найти нетбук после того, когда моя ненаглядная его «приберет».



Ник Вейч
Быстронадуваемый стул/кресло/табурет, для этих бесконечных ожиданий в аэропортах.



Сюзан Линтон
Необходимы выдвижные открывашка и штопор – получится такой швейцарский армейский нетбук.



Шашанк Шарма
Монетоприемник для гостевых пользователей. Деньги пойдут на покупку приличного компьютера.



Маянк Шарма
Нетбуки должны менять законы физики, замедляя время, особенно при выполнении ресурсоемких задач.



Такие маленькие ноутбуки

Полтора-два года назад нетбуки казались той палочкой-выручалочкой, которая вынесет Linux в массы. Все мы помним восторги по поводу Eee PC, радость по случаю появления аналогичных моделей у других производителей... Увы, чуда снова не произошло. Да, про Linux узнало больше людей, но количество не перешло в качество – по крайней мере, в то качество, которого можно было бы ожидать.

Кто-то скажет, что все дело в неудачном выборе дистрибутивов, но мне кажется, причина в не оправдавшей себя бизнес-модели. Производители видели в нетбуках терминалы для мобильного доступа в Интернет, большие MID'ы, где Linux (от Android до Maemo) действительно чувствует себя весьма уверенно. Потребители же разглядели в них маленькие ноутбуки, которые легко носить с собой для эпизодического решения стандартных задач. Ответ: «В нем нельзя запускать ваши программы» на с. 20 все объясняет: если кто-то не хотел видеть Linux на своем ПК, он вряд ли будет пользоваться им на нетбуке.

Проиграл ли Linux очередную битву? Пока что нет. Его преимущества для компьютеров «одной задачи» (хотя бы и работы в Интернете) по-прежнему в силе, и Google уже готовится нанести новый удар, занимаясь разработкой Chrome OS. Другая область, где свободное ПО готово показать себя – это устройства на базе ARM, в каком-то смысле стирающие грань между нетбуком и традиционным планшетом. Да и Windows, как известно, там запустить не так-то просто.

В общем, будущее обещает быть не безоблачным, но интересным.

Валентин Синицын, главный редактор
» info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru
› Адрес редакции: Россия, Санкт-Петербург, Лиговский пр., 50, корп. 15
› Телефон редакции: (812) 309-06-86. Дополнительная информация на с. 104



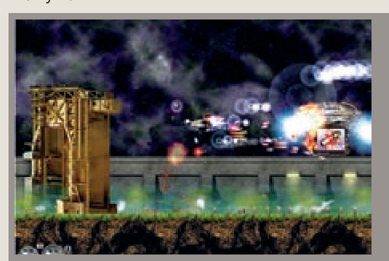
Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры



› **Wiz — вторая крупная попытка создать карманную игровую приставку, предпринятая в Gamepark.**



› **Вернемся в старые добрые деньки: суббота, парк, игровые автоматы, *R-Type* и ZX-Spectrum внутри....**

Мегатест нетбуков



Мы взяли восемь популярных моделей и мучили их, пока они не задымились с. 16

Что за штука...
APML
Сообщите web-сайтам, что вам интересно, и не читайте ерунду с. 38



Профилирование с. 34



Заставьте код выполняться быстрее, но сделайте это правильно.

Люди говорят



«Конечно, главный рецепт победы — это делать хорошие проекты.»

Алексей Балакин раскрывает секрет успеха на Les Trophées du Libre с. 26



Linux Mint 7

Свежее веяние в мире Ubuntu

» **Ulteo** Дружелюбные к пользователю системы выходят на новый уровень

» **Russian Fedora Remix 11** Fedora для отечественных пользователей

» **EduMandriva 2009 Spring LXDE** Образовательный LiveCD со скромными требованиями к ресурсам

...и много чего еще! **с. 96**

Ищите в этом номере...

Постоянные рубрики



› **Наша команда всегда рада подписать вас на LXF. Электронная версия в подарок!**

Учебники





LINUX FORMAT Новости

ГЛАВНОЕ: Нетбуки на ARM и мобильный Linux » Mono: за и против » PostgreSQL 8.4 и VirtualBox 3.0

Нетбуки: Будет ARM – будет Linux



» Рубрику ведет ЕВГЕНИЙ КРЕСТНИКОВ

Наверное, кое-кому тема «мобильного» Linux слегка поднадоела. Но что делать, если это сейчас одно из основных направлений развития свободного ПО и уж точно – самое заметное для простых пользователей.

Android, Moblin, Maemo, Ubuntu Netbook Remix (список можно продолжить) – еще недавно эти названия мало кто знал, не считая некоторого количества энтузиастов. Сейчас реализации встраиваемого Linux у всех на слуху. Но речь пойдет не о них. Что касается ПО – я хочу отметить только тенденцию к универсализации встраиваемых систем на основе GNU/Linux. Разработки, изначально предназначенные для телефонов, пытаются адаптировать для нетбуков, интернет-планшетов и других гаджетов. За примерами далеко ходить не надо – нетбуки под управлением Android должны появиться на рынке уже в этом году (кстати, Android LiveCD для Asus EeePC можно скачать уже сейчас). В общем, не нужно быть пророком, чтобы понять: скоро два направления сольются, и мы получим универсальные встраиваемые ОС для любого типа мобильных устройств. Собственно о программном обеспечении более говорить нечего – стоит упомянуть только о грядущем переходе платформы Maemo с *GTK+* на *Qt*, но эта новость ожидаемая, если учесть что разработкой *Qt* сейчас занимается Nokia.

Перейдем к «железу». Интересных новостей о Linux-телефонах пока нет (выпуск новых моделей не в счет – по ним уже «отстрелялись» новостные порталы). Сегодня мы поговорим о нетбуках. В последнее время производители часто радуют сторонников альтернативных ОС. Во-первых, стоит отметить большое количество анонсов нетбуков на основе архитектуры ARM – к концу 2009 г. они должны появиться в продаже. Разумеется, работать машины будут под управлением Linux. Разработчики ПО готовятся к этому событию, и сейчас активно идет процесс переноса популярных дистрибутивов на ARM (мы уже писали о сборке Ubuntu; кроме того, есть информация о портировании Android, а также недавно была выпущена сборка Slackware для ARM).



› **NVIDIA Tegra серии 600 работает на уровне продвинутых настольных компьютеров.**

Кстати, говоря о нетбуках на ARM, стоит отметить платформу NVIDIA Tegra 600. Она относится к так называемым «системам-на-кристалле» (System-on-Chip, SoC) и содержит ядро ARM-11, графический контроллер и периферию. Разработана Tegra в основном для интернет-планшетов, однако первый нетбук на ее основе должен появиться уже в сентябре (выпустит его компания MobInnova). Судя по пресс-релизу, за $200 покупатель получит устройство с диагональю матрицы 8,9", оснащенное адаптерами Wi-Fi и 3G. Заявленное время работы аккумуляторов – 5–10 часов в режиме просмотра HD-видео, либо до 24-х дней воспроизведения аудиоконтента при закрытом экране.

Еще одно интересное событие (не относящееся к ARM) – компания Nokia планирует начать выпуск нетбуков. Подробности пока неизвестны, однако этот ход вполне ожидаем – финский гигант уже давно выпускает интернет-планшеты, и подобное расширение линейки продуктов компании выглядит логично, особенно если учесть недавние инициативы Nokia в области открытого ПО. Есть и не очень радостные события: представители ASUS заявили о планируемом прекращении поставок нетбуков с предустановленным Linux. Естественно, что ожидаемый многими Eee PC на ARM пока выпускаться не будет. Но все не так страшно – поклонники Eee PC всегда смогут установить на свои машины Linux самостоятельно. А если учесть недавно появившуюся у ASUS процедуру возврата денег за предустановленную версию ОС Windows, платить «налог Microsoft» им, скорее всего, не придется.

> «Идет процесс переноса дистрибутивов на ARM.»

Вообще, если говорить о продвижении Linux в сегмент нетбуков, стоит признать, что оно идет не так быстро, как ожидалось – многие пользователи предпочитают покупать машины с Windows XP либо удалять Linux и устанавливать Windows самостоятельно. Перспективы, конечно, радужные, но с продажами пока туговато. А еще стоит учесть приближающееся начало продаж Windows 7. Наверняка Microsoft выпустит редакцию новой ОС, предназначенную для нетбуков – они не захотят терять этот рынок. Разумеется, «семерка» потеснит Linux куда сильнее, чем устаревшая Windows XP. Так что надежды на серьезное продвижение свободных ОС в этом сегменте я связываю только с платформой ARM – других вариантов нет. Предлагаю дождаться конца года и посмотреть, как будут развиваться события; пока делать прогнозы рановато.

Mono: Быть или не быть?

Вопросы рисков, связанных с использованием Mono, обсуждаются столько лет, сколько существует этот проект.

Недавно спор вновь перешел в активную фазу. Сложно сказать, что стало тому причиной – ситуация вокруг патентов на использованные в Mono технологии не изменилась. Тем не менее, интернет-издание iTWire (www.itwire.com) опубликовало критическую статью, речь в которой шла о невозможности получения лицензий на использование запатентованных технологий. Более того, до сих пор достоверно неизвестно, существуют ли какие-либо патенты на C# и CLI. По крайней мере, корреспондент iTWire не смог получить ответа на свои вопросы ни в ECMA, ни в Microsoft. Журналисты предполагают, что патенты все же существуют, но в Microsoft сознательно затягивают предоставление информации по ним, поскольку позиция компании по этому вопросу еще не выработана. Естественно, после такой публикации возник очередной «спор на пустом месте». Ричард Столлмен [Richard Stallman], до этого никак не обозначавший свое отношение к проекту, подлил масла в огонь, выступив с предостережением против использования Mono для разработки свободного ПО. Позицию основателя FSF можно выразить примерно так: «В системе GNU есть место для всех языков программирования, но следует избегать зависимости свободных программ от потенциально опасных технологий». Впрочем, едва ли это стало для кого-то сюрпризом.

Гораздо интереснее реакция разработчиков нескольких ведущих дистрибутивов. В рассылке fedora-desktop-list появилась информация, что из Fedora 12 (которой не так давно присвоили официальное кодовое имя Constantine) будет исключена базирующаяся на.NET/C# программа *Tomboy* (ее заменят на *Gnote*). Таким образом, Mono будет полностью удален из состава установочного LiveCD Fedora; правда, в репозиториях дистрибутива *Tomboy* и *Mono* останутся. В Debian Team придерживаются сходной политики. Недавно представители проекта опубликовали официальный ответ на заявление Столлмена, в котором сообщили, что Debian не будет включать Mono и основанные на нем программы в установку по умолчанию. На данный момент в Debian есть три метапакета, определяющих состав среды GNOME: *gnome-core*, *gnome-desktop* и *gnome* – минимальная, базовая и полная комплектации. Mono и зависящие от него приложения будут включены только в последнюю. Официальная позиция Технического совета Ubuntu (Ubuntu Technical Board) совершенно другая – Mono и базирующиеся на нем программы входят во многие редакции дистрибутива (в том числе базовую). Исключать из-за гипотетических патентных проблем (опасность которых, по мнению членов Технического совета, сильно преувеличена) полезное ПО из Ubuntu никто не собирается.

> «Позиция практиков отличается от мнения аналитиков.»

Обратите внимание, как отличается позиция практиков от мнения аналитиков: здесь вопрос о возможности использования Mono вообще не возникает – речь идет только о включении Mono-приложений в установку по умолчанию. И это, на мой взгляд, правильно – зачем отказываться от хорошего инструментария, даже если правовая ситуация вокруг него не определена? В случае появления патентных претензий выбросить спорные пакеты из дистрибутива никогда не поздно. С другой стороны, эта правовая неопределенность мешает развитию проекта – использовать Mono для разработки критически важных приложений не рискуют даже в команде GNOME.



› Мигель де Икаса решил раздвоить Mono: теперь его можно будет называть Стерео.

На сегодняшний день все Mono-зависимые приложения могут быть относительно безболезненно убраны из окружения рабочего стола, но есть шанс, что делать это и не придется. Пока верстался номер, компания Microsoft заявила, что стандарты C# и CLI (ECMA 334 и ECMA 335) теперь подпадают под действие Community Promise (Обещание Сообществу), имеющего юридическую силу. Таким образом, корпорация добровольно отказалась от патентного преследования разработчиков. Вопросы, связанные с реализацией прочих запатентованных технологий (таких как ASP.NET, ADO.NET и Windows Forms) пока остаются нерешенными, но они применяются в свободном ПО в очень ограниченных объемах.



› Ричард Столлмен долго хранил молчание по поводу Mono, но его позиция не стала сюрпризом.

Заявление Microsoft стало ответом на запрос вице-президента Novell и ведущего разработчика Mono Мигеля де Икасы [Miguel de Icaza], который в свете новых обстоятельств заявил о разделении проекта на две ветки: реализующую стандарты ECMA и свободную от патентных ограничений, а также «несвободную». Это еще не полная победа, но теперь у противников Mono будет меньше аргументов.

Кстати, далеко не все эксперты считают опасность Mono серьезной. Джо Шилдз [Jo Shields], участник Debian Mono Group, Debian CLI Applications Team, и Debian CLI Libraries Team, написал статью, в которой попытался дать развернутый ответ противникам Mono и рассказать о технических преимуществах этой платформы. Несмотря на то, что г-н Шилдс говорил от своего имени, его точка зрения совпала с точкой зрения Технического совета Ubuntu [Шилдс около года занимается поддержкой Mono в этом дистрибутиве, – *прим. авт.*].

Открытое ПО: Вести с полей

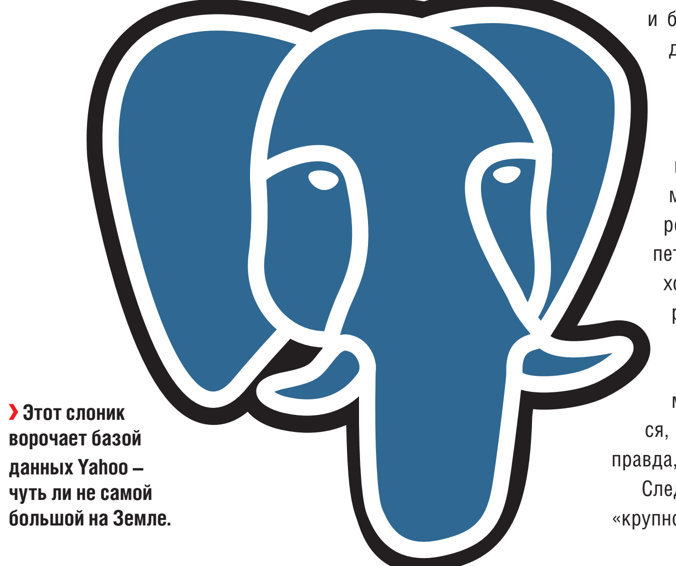
Обычно мы не освещаем в Новостях выход новых версий программных продуктов: соревноваться с онлайн-изданиями – занятие бессмысленное, а для углубленного изучения есть Обзоры. Однако пройти мимо «большого» релиза одной из самых популярных СУБД с открытым кодом было невозможно. Итак, встречайте: PostgreSQL 8.4.

Выпуск состоялся после 16 месяцев разработки. Согласно официальному пресс-релизу, в новую версию вошло 293 улучшения и новых функции. Свои усилия разработчики сосредоточили на оптимизации работы сервера, а также на упрощении процесса администрирования баз данных, написания запросов и программирования. Наконец-то появилась возможность многопоточного восстановления данных, благодаря чему скорость развертывания резервных копий БД увеличилась в 8 раз. Кроме того, администратор теперь сможет назначать права доступа на столбцы таблиц и различные локализации для баз данных.

В общем, новых возможностей «тонкого» администрирования и мониторинга очень много; поговорим лучше о тенденциях развития продукта. Судя по реализуемому функционалу, разработчики *PostgreSQL* явно ориентируются на корпоративный рынок. И с такими темпами развития они скоро смогут на равных поспорить с коммерческими СУБД. Собственно, *PostgreSQL* успешно конкурирует с ними уже сейчас – в качестве примера использования открытого SQL-сервера на высоконагруженных системах можно привести *Skype*. Серверы баз данных последнего, рассчитанные на миллиард пользователей, обслуживают 350 млн абонентов, из которых 10–15 млн постоянно работают в режиме онлайн. Десятки терабайт информации и более 20 тысяч запросов в секунду… если это не enterprise-уровень, то я не знаю, что такое «enterprise». Еще можно рассказать о компании Yahoo, представители которой утверждают, что они создали самую большую базу данных в мире – ее объем составляет около двух петабайт, а число событий в сутки доходит до 24 миллиардов. Если верить этому заявлению, получается, что база данных Yahoo не только самая большая на Земле, но и самая высоконагруженная. Разумеется, компания использует *PostgreSQL* – правда, модифицированный.

> **«Возможностей тонкого администрирования очень много.»**

Следует упомянуть и еще об одном «крупном» релизе – *VirtualBox 3.0.0*. Улучшений множество, но мы отметим только появление Direct3D 8/9 для гостевых Windows-систем, поддержку *OpenGL 2.0* для Windows-, Linux- и Solaris-гостей, а также возможность использования до 32-х виртуальных CPU в гостевом SMP (только для процессоров, поддерживающих технологии VT-x или AMD-v). Еще одна приятная новость – VirtualBox появился в коллекции портов FreeBSD; правда, на момент написания заметки там была только версия 2.2.51, но перенос третьей версии – вопрос времени. LXF



› **Этот слоник ворочает базой данных Yahoo – чуть ли не самой большой на Земле.**

Новости короткой строкой

» Компания Google анонсировала Google Chrome OS – открытую легковесную операционную систему на базе Linux и новой оконной среды, предназначенную для работы с web-приложениями в Google Chrome. Изначально Chrome OS будет нацелена на нетбуки архитектур x86/ARM. Проект независим от Android; первые устройства ожидаются во второй половине 2010 года.

» Запущен *ELRepo* – репозиторий с драйверами различных устройств для RHEL и совместимых дистрибутивов. Цель проекта – продвижение RHEL/Fedora на рабочие станции и настольные компьютеры.

» Стартовал сервис Ksplice Uptrack, который позволит обновлять ядро Ubuntu 9.04 без перезагрузки системы.

» Opera Software открыла спецификацию Scope Transfer Protocol. Он используется для взаимодействия средства отладки *Dragonfly* с браузером.

» Начались продажи Mandriva Linux Powerpack 2009.1, выпущенного специально для российского рынка.

» Открыт доступ к основной ветке разработки KDE 4.4. Примечательно, что это произошло еще до выхода KDE 4.3.

» Исходные тексты Palm WebOS опубликованы по лицензии GPL.

» Проект Sourceforge кардинально изменил дизайн портала.

LINUX FORMAT Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов



АЛЕКСЕЙ ФЕДОРЧУК Его слабости – mass storage, разметка диска и файловые системы.

NILFS выходит из тени

Ядро Linux версии 2.6.30 порадовало нас, в числе прочих новшеств, поддержкой NILFS (New Implementation of a Log-Structured File System) – Новой реализации журнально-структурированной файловой системы. И действительно, среди других ФС последнего поколения, таких как более известные ext4 или btrfs, она выделяется рядом особенностей.

Во-первых, журналирование осуществляется по принципу log-файлов, то есть без перезаписи изменений, а лишь с дополнением журнала изменения состояния файловой системы.

Во-вторых, имеется возможность непрерывного создания снимков, которые могут быть примонтированы в контрольных точках параллельно основной файловой системе – для исправления как системных сбоев, так и пользовательских ошибок.

В-третьих, снимки создаются не путем полного резервирования файловой системы, а лишь записью ее изменений в свободные блоки.

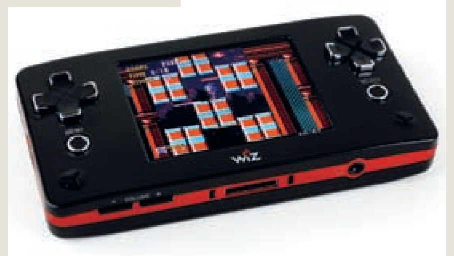
Все эти особенности должны способствовать повышению как надежности, так и производительности. О первой говорить еще рано: NILFS пока что имеет статус экспериментальной системы и не рекомендуется для повсеместного использования. Но ее быстродействие оказывается вполне на уровне подруг-конкуренток – ext4 и btrfs.

С прикидочными результатами можно ознакомиться на моем web-сайте: http://alv.me.

alv@posix.ru

Сегодня мы рассматриваем:

GP2X Wiz с. 8



› Неудобный джойстик прежнего GP2X сменился на дирекционную площадку.

Jets'n'Guns с. 11



› Готовьтесь: когда Большой адронный коллайдер наконец-то починят и запустят, мир может потерять одно измерение.

Наш вердикт: Пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибалльной шкале (0 – низшая оценка, 10 – высшая). Как правило, учитываются функциональность, производительность, простота использования и цена, а для бесплатных программ – еще и документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.



Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но если разработчики рекомендуют *Autopackage*, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчики: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Оправданность цены	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

GP2X Wiz

Взор **Саймона Пикстока** туманится от ностальгии при виде портативной игровой платформы – эмулятора аркад.

Вкратце

» **Портативное игровое устройство на базе Linux и открытого ПО для ненаигравшихся взрослых.**

Спецификация

» **Процессор** 533 МГц
» **Экран** OLED
» **ОЗУ** 1 ГБ
» **Поддерживаемые форматы** MPEG4, DIVX, XVID, JPEG, BMP, GIF, PNG, WAV, OGG, TXT, Flash 8.0

Все любят порой обратиться к ретро, и GP2X Wiz поможет вспомнить былое. Это модифицированная GP2, оснащенная сенсорным экраном под стилус, процессором ARM на 533 Мгц и слотом для карты памяти с играми. Но самое главное – устройство работает на Linux. Wiz соединяется с компьютером через USB-кабель (приобретается отдельно), который, помимо прочего, служит для подзарядки. После подключения вас спросят о том, какую память использовать для передачи: встроенную или SSD-карту. Выбранное хранилище помещается на рабочий стол в виде съемного накопителя, что обеспечивает обмен файлами.

OLED-экран устройства чистый и яркий, а интерфейс состоит всего из шести значков: SD-карта, встроенные игры, флэш-игры, развлечения, модуль запуска и настройка. Встроенные игры – просто клоны старой классики типа *Snake*, *Tower Defence* и т. п. Есть пять флэш-игр, но, за незнанием корейского, нам не удалось расшифровать их названия. Впрочем, беда невелика: это игры для тренировки мозга, и в них нужно сопоставлять символы или заниматься простой арифметикой. В разделе развлечений можно запускать аудио- и видеофайлы и просматривать изображения.

> «Wiz задевает в вас все струнки первобытного хакера.»

Игры требуют пользования сенсорным экраном и стилусом. Правда, чувствительность экрана недостаточна: чтобы «достучаться» до него, придется ткнуть не раз. В итоге консоль установила наш «умственный возраст» как 61 год – ну и ну! Углубленное исследование показало, что на экране сбито расположение активных зон. Повторной калибровкой мы снизили одряхление ума сразу на 20 лет – както лучше!

› **Как приятно вспомнить детство, вновь окунувшись в атмосферу любимой игры… только, пожалуйста, не нарушайте закон!**

Ностальгия

На Wiz уже портировано несколько консольных эмуляторов – например, Megadrive (Genesis) и SNES; вскорости ожидается портирование *Mame* и пр. Однако обрести их не удастся без розыскных мероприятий: на английском сайте Wiz пока нет рабочих ссылок. Материалы содержатся в основном на форумах, требующих регистрации, а также на сайтах фанатов.

К счастью, у Майка уцелела коллекция ROM, и вскоре мы уже оттягивались с *Sonic the Hedgehog* и *Road Rash* на эмуляторе Genesis. Ка-айф… Правда, тут выявился еще один просчет дизайна: левый манипулятор имеет традиционную конструкцию, а правый представляет собой четыре отдельных кнопки, расположенные крестом. Так вот, эти кнопки слишком малы и слишком притиснуты друг к другу, очень неудобно. Кроме того, при загрузке каждого нового уровня громкость звучания выставляется на максимум: это неприятно, если вы находитесь в общественном месте (а при использовании наушников еще и больно).

Отметим также, что найти нужный эмулятор, скопировать его куда надо, заставить работать, а затем отыскать ROM для загрузки – тяжелое испытание. Wiz явно рассчитан на старые консольные игры, но ведь ROM защищены авторскими правами, поэтому добыть их далеко не просто (да и легальность применения сомнительна). Короче, наше путешествие по аллее воспоминаний больше напоминало штурм Эвереста по северной стене, чем обещанную прогулку в парке.

Wiz задевает в вас все струнки первобытного хакера, хотя и через хромые кнопки, и это отличная игрушка для любителей полного контроля над игровым процессом без перепрошивки стандартной консоли. Если же вы игрок от случая к случаю и не намерены тратить полжизни на поиски левого контента, сидите-ка лучше со своей PSP или DS. LXF



› **GP2X Wiz помещается в кармане брюк — но не забудьте вынуть оттуда ключи, а потом уж садитесь.**

LINUX FORMAT Вердикт

GP2X Wiz

Разработчик: Green Park Holdings
Сайт: www.globalgph.com
Цена: около $250

Функциональность	8/10
Производительность	8/10
Простота использования	7/10
Оправданность цены	8/10

» Чтобы добиться оптимальных результатов, придется изрядно попотеть.



7/10

LINUX FORMAT на DVD

Kdenlive 0.7.3

В наше время не иметь редактора видео – дурной тон. **Энди Ченнел** проверяет, как смотрится Linux на фоне iMovie и Windows Movie Maker…

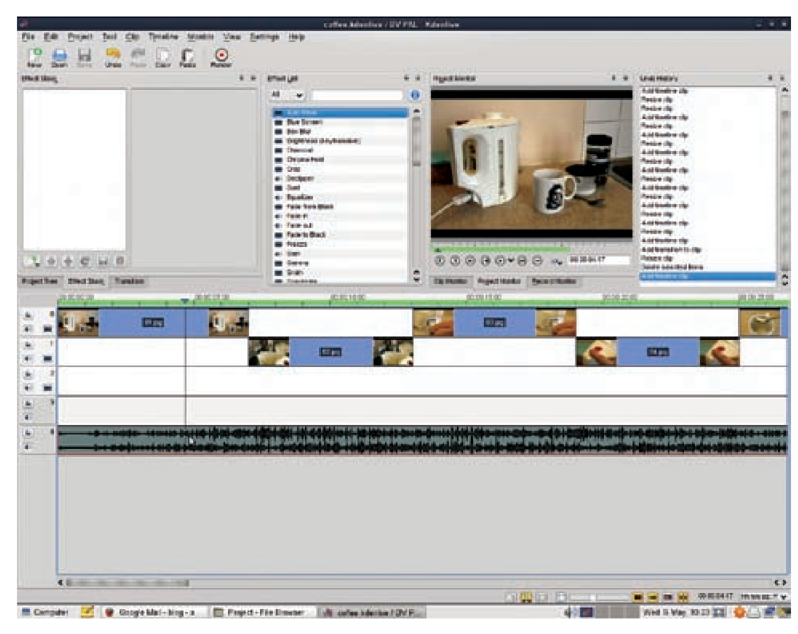
Вкратце

» Комплект для традиционного нелинейного редактирования видео/аудио. См. также: *PiTiVi* **и** *Kino*.

Видео всё глубже проникает в нашу жизнь, а вот видеоредакторы для Linux, похоже, отстают от подобных приложений для других ОС. Бесспорно, *Kino* хорош – но освоить его не так-то легко, а *PiTiVi* еще сыроват для постоянного использования. И тут, словно луч света в темном царстве, появляется *Kdenlive* – отважный новичок с претензиями на популярность *Final Cut* и *Premiere*.

Kdenlive – это нелинейный видеоредактор, использующий знакомый подход к работе с видеоизображением: временную шкалу. Добавление и подрезка клипов тоже знакомо по другим приложениям, как и управление медиа-контентом.

При первом запуске экран перегружен элементами, но их можно закрывать, открывать и масштабировать, в соответствии с выполняемой работой. Например, при захвате видео мы закрыли окна эффектов, переходов и временной шкалы и сконцентрировались на самом процессе; затем, в ходе грубой подгонки, вернули временную шкалу с монитором клипов назад.



› ***Kdenlive*** **справится со всеми аудио- и видеодорожками, которые «вывезут» ваши процессор, память и диски.**

> «При такой гибкости легко сосредоточиться на конкретной задаче.»

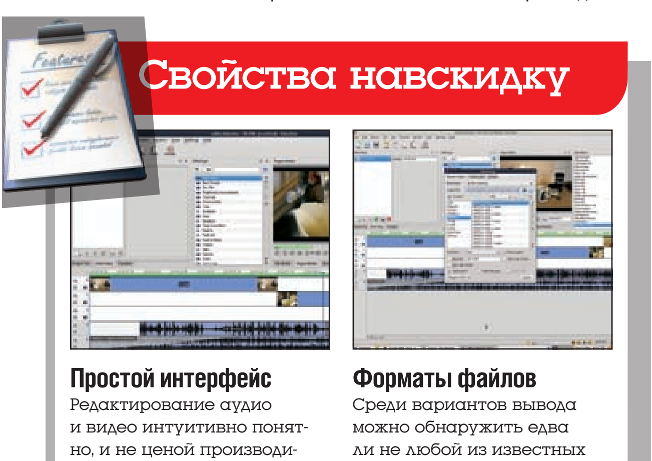
При подобной гибкости, особенно в сочетании с полноэкранным режимом (Ctrl+Shift+F), легко полностью сосредоточиться на конкретной задаче. Логично было бы предположить пригодность *Kdenlive* к работе на нескольких мониторах. Однако в таком режиме (даже на одном мониторе) в ходе наших тестов программа постоянно «сыпалась». В остальном приложение было на удивление стабильно.

Кадры решают все

После импорта клипов работа с фильмом сводится к перетаскиванию отрывков на временную шкалу и подгонке их инструментом Razor [Лезвие]. Процесс интуитивно ясен; отдельного упоминания стоит ползунок Zoom [Масштабирование], облегчающий переход между общим и детальным видами. Удобно и контекстное меню по правому щелчку: обращаться к основной строке меню приходится довольно редко.

Подогнав клипы к временной шкале, оформляйте фильм встроенными эффектами. Они тоже добавляются на временную шкалу перетаскиванием, а объединить их или перетасовать их порядок поможет окно Effects Stack [Группировка эффектов]. Эффекты логично подразделяются на видео, аудио и «особые», причем для каждого предусмотрена группа настраиваемых параметров. Типов перехода всего четыре – вроде и немного; но, например, переходы типа Luma снабжены массой различных режимов растворения, затухания или вытеснения; тип Compositor обеспечивает разделение кадров, добавление экранной графики, и т.д. Все эти эффекты можно применять к ключевым кадрам, получая весьма изощренные композиции. Непосредственный рендеринг эффектов, включая сгруппированные, работает безупречно.

Короче, блестящее приложение для работы с видео. Освоить его недолго, пользоваться несложно, и программа необычайно мощная. Отличный выбор и для профессионального видеомонтажера, и для любителя, записывающего домашнее видео на DVD. LXF



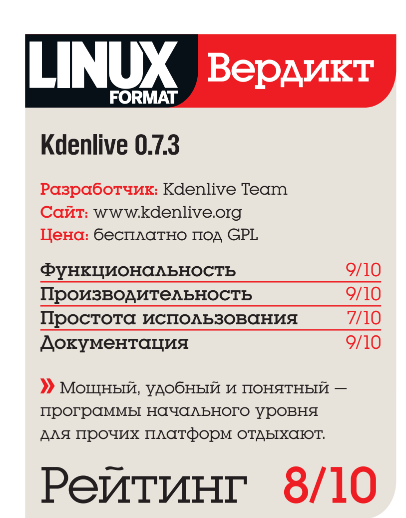
Свойства навскидку

Простой интерфейс

Редактирование аудио и видео интуитивно понятно, и не ценой производительности.

Форматы файлов

Среди вариантов вывода можно обнаружить едва ли не любой из известных форматов.



LINUX FORMAT Вердикт

Kdenlive 0.7.3

Разработчик: Kdenlive Team
Сайт: www.kdenlive.org
Цена: бесплатно под GPL

Функциональность	9/10
Производительность	9/10
Простота использования	7/10
Документация	9/10

» Мощный, удобный и понятный — программы начального уровня для прочих платформ отдыхают.

Рейтинг 8/10

Ulteo OVD 1.0

Очередное детище Гаэля Дюваля, Open Virtual Desktop – виртуальный рабочий стол прямо в браузере. **Маянк Шарма** проверяет, достоин ли новый «член семьи» своей родословной.

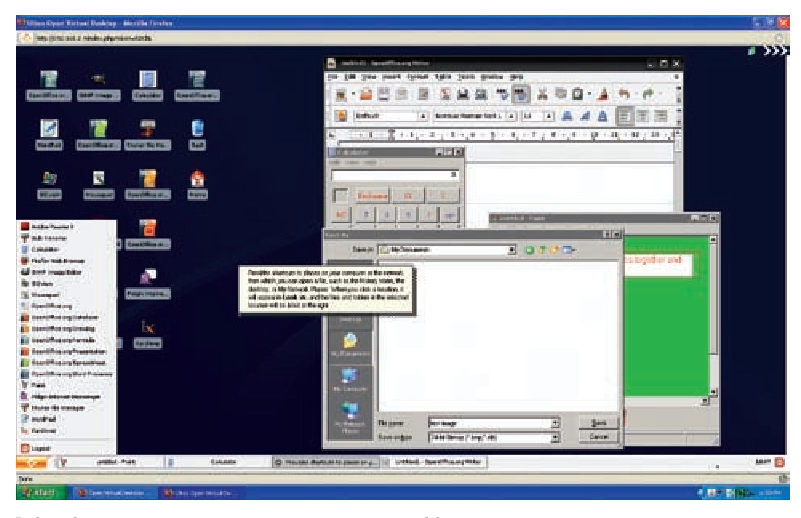
Вкратце

» **Виртуальный рабочий стол, поддерживающий и Linux-, и Windows-приложения. Среди проприетарных альтернатив—***VMware View* **и** *Sun VDI***.**

Гаэль Дюваль [Gael Duval] всегда был трудягой. Человек, подаривший нам дружелюбный дистрибутив Mandrake (позже – Mandriva), был уволен из основанной им самим компании в 2006 году, и сразу же занялся проектом под названием Ulteo.

Сейчас перед нами первый стабильный продукт семейства Ulteo. *Open Virtual Desktop* (*OVD*) – это система, использующая одну или две серверных машины (одна для Linux-, другая для Windows-приложений), чтобы обслужить несколько тонких клиентов через виртуальный рабочий стол в окне браузера. Особая крутизна в том, что работать можно на любой системе-хосте.

По сути, *OVD* не такая уж революция – система построена на базе инфраструктуры Virtual Desktop Infrastructure (VDI); это просто вывод сугубо промышленной технологии «в массы». Примерно то же Гаэль в своё время проделал с Mandrake Linux.



› *OVD* **безопасно доставляет приложения через SSH-туннели и умеет, согласно документации, обслуживать одновременно 20 пользователей на 1-ГБ ОЗУ многоядерной машины.**

Развяжем рюкзачок

Два основных компонента, составляющих Linux-половину *OVD* – сервер приложений и менеджер сеансов; оба открыты и бесплатны. Менеджер сеансов – это web-консоль для настройки клиентских машин, а сервер приложений (как нетрудно догадаться) обслуживает Linux-приложения. Windows-приложения обслуживаются Windows-агентом; его надо установить на отдельный экземпляр Windows Server 2003 с настроенной службой терминалов.

> «OVD несет сугубо промышленные технологии в массы».

Звучит пугающе, но на практике всё просто. Для основных Linux-компонентов есть двоичные пакеты, а процесс установки хорошо документирован. Сервер и менеджер сеансов можно поставить на одну и ту же машину, а специальный DVD позволит установить весь комплект на измененную копию Ubuntu. Windows-половина устанавливается по желанию. Замечательно то, что *OVD* можно сцепить с сервером службы каталогов – поддерживаются и Active Directory, и LDAP. Вся работа делается через серверы приложений, и *OVD* можно применить в системе «тонких клиентов», храня файлы на сервере CIFS.

Интерфейс

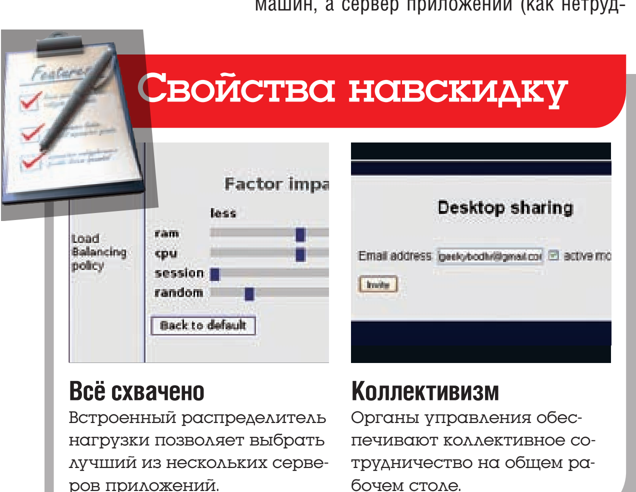
OVD поставляется с пакетом готовых Linux-приложений, включая *OpenOffice.org*, *Firefox*, *Adobe Reader* и *GIMP*. Можно установить и другое ПО, указав название в менеджере сеансов: необходимая программа будет получена из специального репозитория через *apt-get*. Приложения можно закреплять за отдельными пользователями, а также объединять по группам. Перечень Windows-приложений контролируется через службы терминалов из-под Windows Server.

Администрирование также осуществляется с помощью менеджера сеансов, давая возможность устанавливать таймауты, скрывать или отображать значки рабочего стола, переназначать клавиши и задавать язык интерфейса.

В ходе наших тестов приложение без труда справлялось с обслуживанием двоих пользователей на виртуальной машине со 128 МБ виртуализированного ОЗУ (двухъядерный хост).

В будущем разработчики Ulteo обещают функцию запуска приложений по нажатию гиперссылки, без разворачивания всего рабочего стола. Пока же остаётся наблюдать, потянутся ли к *OVD* пользователи. LXF

Свойства навскидку



Всё схвачено
Встроенный распределитель нагрузки позволяет выбрать лучший из нескольких серверов приложений.

Коллективизм
Органы управления обеспечивают коллективное сотрудничество на общем рабочем столе.

Linux Format Вердикт

Ulteo Open Virtual Desktop

Разработчик: Ulteo
Сайт: www.ulteo.com
Цена: бесплатно под GPL

Функциональность	9/10
Производительность	9/10
Простота использования	9/10
Документация	8/10

» OVD от Ulteo – система тонких клиентов, идеальная для корпоративной работы при смешении платформ.

Рейтинг 10/10

Jets'n'Guns Gold

Быстрее ударь, срочно прокачайся – и снова бей, бомби, круши! **Ник Вейч** снимает стресс от супермаркета при помощи старомодного развлечения.

Вкратце

» **Стрелялка с бесподобной графикой. См. также:** *Project Starfighter*.

При мыслях об играх, которые непременно следует портировать на Linux, стрелялки с боковой прокруткой приходят на ум далеко не первыми. Они достигли вершин в *R-Type*, затмившей аркады конца 80-х, и хотя были ориентированы на ящики, глотающие монеты в обмен на адреналин, всё же манят и игроков с настольными ПК. Прежде чем браться за их доработку, неплохо бы выбрать наилучший вариант.

В *Jets'n'Guns* налицо все классические элементы жанра: бодрящий саундтрек, настраивающий на агрессивный лад; простая система управления; элемент стратегии (ну, или тень намека на него); невообразимый арсенал вооружения, а также отличная, сверхнасыщенная графика. Даже сюжет есть!.. Ну, это я погорячился, но оформленные в умеренно-комическом стиле панели в переходах между уровнями придают игре известную связность.

Стреляя во всё, что шевелится, зарабатывайте деньги и покупайте еще более разрушительное оружие – вот и вся сказка. Кстати, вам дается несколько единиц

> «На 42-х уровнях немало тактических вариаций.»

оружия – и это один из шагов *J'n'G* вперед по сравнению с прототипами. Пушки это, ракеты или бомбы, зависит от типа носителя (его тоже можно обновить).



› **Потом придется освоить модифицированный биплан и стрелять. Много стрелять.**



› **Арсенал набит оружием, и на раскачку времени нет.**

Можно составлять комплекты вооружения, сохранять их и переключаться между разными комплектами во время игры – приятное добавление. Да и полезное, если приходится иметь дело то с наземными целями, то с эскадрильями самолетов, то с бронетехникой.

Идем вразнос

На 42-х уровнях разрушения немало тактических вариаций, но все они сводятся к повышению эффективности стрельбы. Изредка попадаются несложные загадки, связанные со вскрытием оружейных складов: секреты, которые нужно разгадывать, бонусные уровни, которые нужно вскрывать, кровавые сцены, на которые стоит посмотреть – и всё это на фоне одного из лучших визуальных оформлений для Linux (необходима видеокарта с поддержкой OpenGL). Несмотря на скромное разрешение (всего 800×600), графика впечатляет.

Единственное реальное неудобство (кроме скуки, усиливающейся с ростом вооруженности) – это управление. Реакция мыши отнюдь не мгновенная, а попытки настроить кнопки на геймпаде оказались бесплодными, так как ввод направления не воспринимался (конечно, джойстики в Linux – это отдельная песня); не лучше обстоит дело и с некоторыми другими клавишами. Пользователям Fedora, чтобы заставить игру работать, придется шаманить с командной строкой, так как программа едва ли сможет сама найти библиотеки OpenGL (хотя под Ubuntu на тех же машинах всё идет как надо!).

Трехчасовой (как минимум) блестящий спектакль, щедро пересыпанный юмором, по стандартам коммерческих игр вполне стоит запрошенных денег. Да, вы уже видели все это раньше – но, наверное, не на столь высоком уровне. LXF



LINUX FORMAT Вердикт

Jets'n'Guns Gold

Разработчик: Rake in Grass/ LGP
Сайт: www.linuxgamepublishing.com
Цена: £15

Функциональность	6/10
Производительность	7/10
Простота использования	7/10
Оправданность цены	5/10

» Экстремально крутая стрелялка с боковой прокруткой.

Рейтинг 7/10



OpenOffice.org 3.1

Несмотря на туманность будущего, настоящее известного офисного пакета выглядит весьма неплохо. **Энди Ченнел** оценивает его недавние достижения…

Вкратце

» **Флагманский офисный пакет для Linux. См. также:** *KOffice*, *Abiword/Gnumeric* **и Google Docs.**

Презрев слухи о своей кончине, забурлившие, когда Oracle приобрела Sun, проект *OpenOffice.org* выпустил версию 3.1 одноименного офисного пакета. Удалось, наконец, повысить плавность отрисовки и немного, но чувствительно сократить время загрузки.

Медленный старт был слабым местом проекта с момента его основания, и данная версия не исправила положение. В наших тестах *OOo Writer* – избранный за особую востребованность пользователями – стартовал из «холодного» состояния чуть быстрее семи секунд на машине с 3-гигагерцевым Pentium 4 и 1,5 ГБ ОЗУ. Вроде и неплохо, но *Word XP* (на той же машине под *Wine*) на две секунды его опередил. При повторной загрузке счет был четыре и две секунды соответственно: опять поражение *OOo*. Обоих, правда, послал в нокдаун *AbiWord*, вынырнувший из «холодного» состояния всего за 1,43 секунды. Впрочем, на более мощных машинах разрыв между *OOo* и *MS Office* гораздо меньше, и обе программы запускаются из исходного положения секунды за четыре.

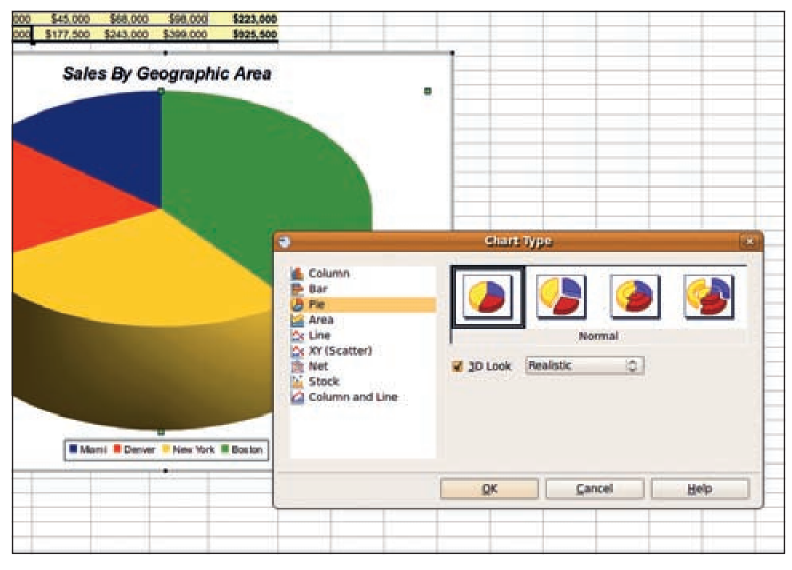
> «Не было задержек при выделении, редактировании и перемещении.»

Что новенького?

Сокращение времени загрузки, конечно, уже повод для хвастовства, но гораздо интереснее поведение приложения в работе. Запущенный *OOo* занимает около 19,5 МБ памяти, хотя этот объем неизбежно растет при работе с документами. Для проверки в условиях интенсивной нагрузки мы загрузили крупный документ *Word* с несколькими иллюстрациями высокого разрешения, а также комментариями и исправлениями от разных авторов. Приложение держалось стойко – не было задержек ни при выделении и редактировании текста, ни при перемещении иллюстрации в пределах страницы, а панорамирование и масштабирование проходили как по маслу. Использование памяти увеличилось до 75 МБ, но загрузка процессора оставалась весьма умеренной, достигая максимума в 10 % при копировании и вставке больших объемов текста.

Для слабых машин загрузка процессора и использование памяти при работе с крупными документами, возможно, станут проблемой, но теперь, по крайней мере, нет гнетущих пауз между исчезновением объекта из одного места и появлением его в другом – а это явное улучшение по сравнению с предыдущими версиями.

В результате упорной работы появилось еще одно изменение, влияющее на весь пакет: сглаживание контуров объектов, добавляемых на страницы. Если в прежних версиях края изображений казались зазубренными, то теперь векторные рисунки, созданные в *OOo*, отображаются аккуратно, невзирая на их масштаб. Это здорово помогает при работе со шрифтовыми эффектами (они же Fontwork) во *Writer* или при «выдавливании» в *Impress* для придания объектам глубины. Разумеется, качество растровой графики от этого не изменилось, но если вы работаете в *OOo* с презентациями или простым дизайном, то визуально ваша работа заметно выиграет. Качество схем и графиков, получаемых с помощью *Calc*, тоже повысилось: теперь они выглядят превосходно.



» **Вероятно, обычных пользователей сильнее всего впечатлит улучшенное, сглаженное отображение графики (например, схем и шрифтов).**

Добавьте контент

Кроме общего визуального улучшения, переработано перемещение объектов в реальном времени, что упрощает компоновку документов – теперь рисунки и иллюстрации прорисовываются во время «перетаскивания» непрерывно, а не обозначаются пунктирными линиями. Изменение не радикальное, но порадует любителей точности – да и выглядит приятно.

Версия, которую мы загрузили для Ubuntu, снабжена неплохим набором шаблонов для презентаций. Они, конечно, уступают по качеству аналогам из *Keynote* от Apple или *PowerPoint 2007*, но как отправная точка вполне годятся; к тому же,

Свойства навскидку



Гладкая графика
Благодаря встроенному сглаживанию краев объектов они выглядят более четко – отлично для презентаций.

Дадим отпор
Новая система работы с комментариями превращает просмотр документа в плодотворное обсуждение.

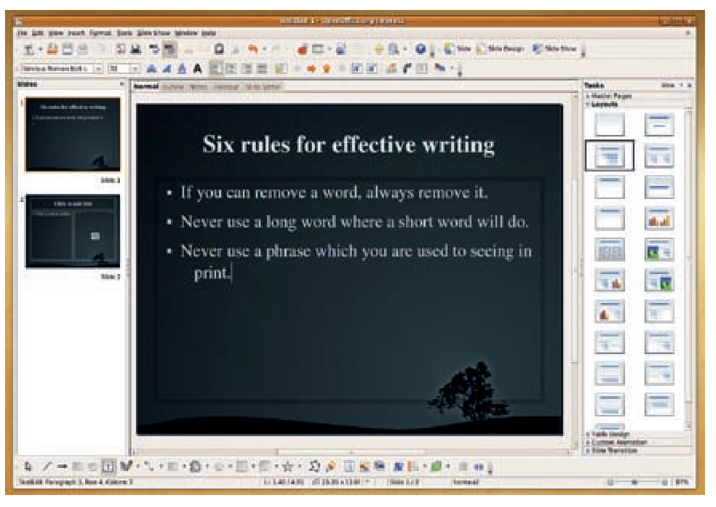
если «штатный» набор недостаточен, в Интернете можно найти немало шаблонов, созданных пользователями. *Impress* наконец-то обзавелся ползунком масштабирования (у *Writer* он появился в версии 3), что упрощает навигацию по сложным, насыщенным слайдам.

Работай играючи

OpenOffice.org всегда предоставлял свойства документа, через которые можно было внедрять в файл различные строки текста. Версия 3.1 пошла дальше: в ней появилась функция Custom Properties [Специальные свойства]. Она позволяет добавлять к документу метаданные, значительно облегчающие обращение с документом. Например, при подготовке документов для нескольких заказчиков в Custom Properties можно внести сведения о клиентах, об ответственных лицах, о сроках и порядке сдачи работы. Система оснащена готовым набором полей, но для добавления собственного поля достаточно написать название свойства, определить тип контента (текст, число, дата и т. п.) и ввести необходимые данные. Новшество пригодится не всем, но для тех, кто издает множество документов или работает в коллективе, это очень удобно.

Усовершенствованы и другие инструменты совместной работы. В версии 3 *Writer* получил систему заметок на полях, подобную используемой в *Word*. Теперь можно комментировать комментарии – это позволяет составителям документа «парировать» замечания коллег, превращая процесс подготовки документа в подобие дискуссии. В наших тестах *OOo* исправно импортировал комментарии и отслеживал изменения в документах *Word*, а потом и экспортировал их.

Те, кому часто приходится работать над документами совместно, оценят также преимущества новой системы блокировок. Если один из составителей взялся за документ, другие пользователи не смогут открыть и изменить файл даже по сети. Раньше данная функция действовала только при работе всех пользователей на одинаковых ОС; теперь документы надежно блокируются на любых платформах.



› **Новая версия предоставляет симпатичные шаблоны, вдобавок ничто не мешает создать свой собственный.**

Грэм сказал...

» «Я не особо в восторге от последней версии. По-моему, прогресс *OOo* слишком долго ограничивался потугами догнать *MS Office*.»

С каждой версией *OOo* импорт и экспорт файлов популярных форматов, типа DOC и XLS, постоянно совершенствуется. В русле этой тенденции, *OOo 3.1* улучшил поддержку новейших форматов на основе XML, хотя она покамест далека от совершенства. В ходе наших тестов открытие большинства файлов DOCX, XLSX и PPTX приводило либо к зависанию, либо к неполному отображению данных. Однако файлы почти всех «старых» форматов открываются без проблем – в общем, если вы взаимодействуете с пользователями *MSO*, посоветуйте им быть «консервативнее».

Что в будущем?

После поглощения Sun громадой Oracle возникло подозрение, что дни *OpenOffice.org* сочтены. Бесспорно, проект многим обязан массовому участию программистов Sun, но такая поддержка предполагает неминуемое замедление процесса разработки – оборотная сторона корпоративности. Короче, еще неизвестно, кому повезло.

Майкл Микс [Michael Meeks] из Novell и другие программисты годами твердили о том, что короткий поводок Sun тянет проект назад, а новые полезные функции или поправки вязнут в череде согласований и устаревают, так и не попав в стабильный вариант пакета. Если этот «ошейник» ослабнет – откроется прекрасная возможность для Red Hat, Novell, Canonical, IBM (в успехе проекта кровно заинтересованы все) и для «домашних хакеров» сделать шаг вперед и вдохнуть новую жизнь в разработку *OpenOffice.org*.

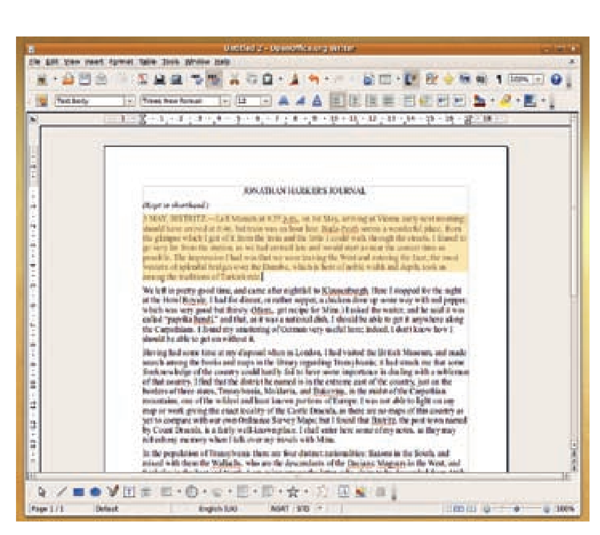
Впереди возможны и осложнения, но *OpenOffice.org* – знамя открытого ПО, и весьма сомнительно, что крупные игроки допустят его исчезновение. А если процесс разработки замедлится из-за новой волны согласований, на это время мы, по крайней мере, уже обзавелись высококачественным ПО.

Все-таки № 1

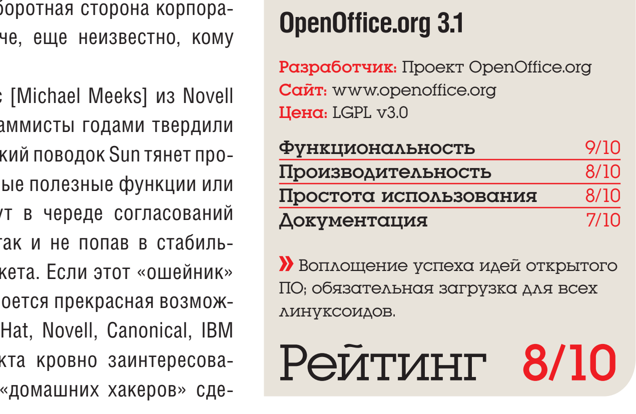
Итак, пускай с *AbiWord* экономится пара секунд на загрузке, а с *KOffice* ощущается причастность к общему делу – в поисках открытого пакета для офисной работы вы первым делом обратитесь к *OpenOffice.org*. Его традиционно сильные стороны – совместимость с форматами фактического лидера *MS Office* и высокое качество получаемых документов. Мы приветствуем увеличение скорости, улучшения в области работы с графикой, инструменты совместной работы, обновление системы отслеживания комментариев и правок – все это облегчает пользователям *OOo* взаимодействие друг с другом и с пользователями других пакетов.

Самое слабое место офиса – база данных: ее стабильность оставляет желать лучшего. Она выполняет полезную работу – предоставляет источники данных для других компонентов пакета; но сама база, пожалуй, чересчур сложна для обычного пользователя.

Наконец, единство функциональности, обеспечиваемое *OOo* на трех основных платформах, обеспечивает безболезненный переход с Windows на Linux, ведь свойства ПО при смене ОС не меняются. В итоге, *OOo 3.1* настоятельно рекомендуется. LXF



› **Цветное выделение удобнее и выглядит куда лучше, чем прежний инверсный эффект.**



LINUX FORMAT **Вердикт**

OpenOffice.org 3.1

Разработчик: Проект OpenOffice.org
Сайт: www.openoffice.org
Цена: LGPL v3.0

Функциональность	9/10
Производительность	8/10
Простота использования	8/10
Документация	7/10

» Воплощение успеха идей открытого ПО; обязательная загрузка для всех линуксоидов.

Рейтинг 8/10

EnergyXT 2.5

Модульная музыкальная композиция – оправдание записи сразу по нескольку треков для серийного музыкального маньяка **Грэма Моррисона**.

Вкратце

» **Модульное музыкальное приложение с поударной разбивкой, эффектами реального времени и виртуальными синтезаторами. См. также:** *Ardour*, *Renoise 2.0* **или** *Rosegarden*.

В обзоре *EnergyXT* (**LXF94**) мы восхищались мощным потенциалом этого кросс-платформенного комплекса с модульной архитектурой. Посудите сами: вместо подражания стандартным окнам микшерного пульта и аранжировки, *EnergyXT* позволяет удобным для вас образом связать композиционные инструменты, звуковые генераторы, аудиоэффекты и модули записи.

Версия 2.5 построена на том же принципе, но позаботилась о ряде удобств. В комплект входят MIDI-секвенсор, многодорожечный рекордер и гитарный проект, поэтому необходимость в «грязной» работе при создании или записи мелодии отпадает. Например, по щелчку на шаблоне Drum And Bass [Барабаны и бас] загрузится проект с комплектом ударных инструментов и басовым синтезатором (по умолчанию – с окнами нотной записи для обоих источников звука). После этого остается лишь щелкать по нотам да нажимать кнопку Play [Играть] – трек лепится за сущие секунды.

> «Нарезка и растяжка музыкальных циклов – автоматом на лету.»

Теперь можно вообще обойтись без ручной компоновки в окне модулирования. Эффекты, источники звука и даже секвенсоры перетаскиваются с панели ресурсов в главное окно – примерно так, как в *Ableton Live* на OS X и Windows. Еще одна аналогия: при смене темпа возможна автоматическая нарезка и растяжка музыкальных циклов в реальном времени. Сэмплы, MIDI-фразы и барабанные пассажи запросто перетаскиваются в окно секвенсора и даже накладываются друг на друга, образуя в итоге затейливые музыкальные узоры. Блоки нот можно сдвигать вправо и влево, а также виртуально переносить между различными модулями – быстро и эффективно.

› **Плавная прокрутка, увеличение, копирование блоков и редактирование — все это умеет всеобъемлющее окно секвенсора *EnergyXT*.**

Полоса информации

Информационная полоска окна микшера («Channel strip») теперь отображается еще и рядом с дорожками, что позволяет корректировать спектр и добавлять эффекты. Правда, не все работает безупречно. Например, если перетащить модуль Arpeggio на дорожку синтезатора, модуль-то добавится, а вот увязывать ноты между секвенсором и синтезатором придется вручную.

Однако более всего приложению недостает поддержки LADSPA и DSSI для «родных» аудиоэффектов и виртуальных инструментов Linux. Вместо этого вы ограничены набором эффектов, входящих в комплект поставки (хотя заставить работать парочку бесплатных VST-инструментов, скомпилированных для Linux, все-таки можно). Смягчают горе несколько новых эффектов, разработанных для версии 2.5 (например, Bit Crusher с ретро-звучанием, фантастический многорежимный фильтр, компрессор и гитарный усилитель).

Встроенный синтезатор звучит превосходно и способен создать любой классический звук, хотя автоматизация работы с параметрами вроде частоты отсечки фильтра в окне секвенсора малоудобна: сначала нужно назначить отсечку как управляющий параметр MIDI, а затем редактировать его в окне секвенсора двойным щелчком на каждом уровне квантования в последовательности. Было бы гораздо проще выбирать параметр из выпадающего списка, а затем редактировать инструментом Pencil [Карандаш]. Новый синтезатор и сэмплер ударных разворачивает запись звуков в окне, а если у вас установлены библиотеки *Lame*, то можно смикшировать проект в MP3. И, несмотря на досадные пробелы в поддержке эффектов и инструментов, трудно назвать приложение для Linux, столь же гибкое и эффективное в использовании. LXF





Поударная разбивка

Цикл можно динамически подогнать под темп канала, не меняя тональности.



Окно микширования

Новый «пульт микширования» позволит перетащить эффекты на каждый канал.



Вердикт

EnergyXT 2.5

Разработчик: XT Software
Сайт: www.energy-xt.com
Цена: €59

Функциональность	8/10
Производительность	9/10
Простота использования	7/10
Оправданность цены	9/10

» Быстрый, мощный и эффективный. С поддержкой «родных» эффектов и синтезаторов Linux станет еще интереснее.





Выбираем идеальный нетбук

Мы заперли Майка Сондерса в комнате с восемью нетбуками, недельным пайком кислорода и заданием: найти идеальную машину для каждой категории пользователей.

Здесь, в *Linux Format*, нет упертых ненавистников Microsoft, но мы всегда хихикаем, вспоминая прогноз Билла Гейтса [Bill Gates] о планшетных компьютерах, сделанный им на Comdex в 2001 году. «Предсказываю, что в течение следующих пяти лет они станут самыми популярными видами ПК, продаваемыми в Америке», заявил тогда самый знаменитый миллиардер в мире ИТ, и здорово промахнулся. Планшеты все еще считаются экзотикой и ограничены несколькими нишевыми сегментами рынка.

Зато растущий рынок нетбуков преподнес всем сюрприз. Когда Asus выпустил Eee PC 701, многие пуристы сочли его детской игрушкой, однако его скудного оборудования оказалось с лихвой достаточно для задач нашей повседневной жизни. Беглый обзор web, офисная работа, пасьянсик в поезде – Eee делал свою работу хорошо, особенно при ценовой отметке в несколько сот долларов, и после него пошли модели с бо́льшими экранами и клавиатурами, дабы умерить жалобы на неудобства первой версии.

Крупные игроки, такие как Dell, Lenovo и Toshiba, тоже ввязались в гонку, и хотя их компьютеры в общем укладываются в низкобюджетную категорию (менее $500), тем, у кого на счету каждая копейка, важен выбор правильного экземпляра. Все они предоставляют доступ к Интернет через Wi-Fi и выполняют офисные задачи, но, как мы увидим, могут сильно отличаться в ключевых аспектах.

Наши критерии оценок

Мы сгребли все нетбуки, которые смогли достать, чтобы тест был всеобъемлющим. Мы рассматривали:

» **Производительность** Все нетбуки, кроме одного, основаны на 1,6-ГГц CPU Intel Atom и графическом чипе 945GME. Однако другие компоненты, особенно устройства хранения данных и беспроводные сетевые карты, тоже играют важную роль. Мы обращали внимание и на эти аспекты.

» **Удобство** Это самый важный момент. Неважно, насколько хорошо выглядит нетбук, если его клавиатура очень тесная или сенсорная площадка никуда не годится.

» **Качество сборки** Нетбук не для того, чтобы держать его в вате. Его таскают в сумке и используют где попало, не боясь стукнуть раз-другой.

Мы знаем, что многие постоянные пользователи Linux предпочитают устанавливать собственный дистрибутив, и чтобы наши испытания были справедливыми, мы поставили Ubuntu 9.04 Netbook Remix на каждую машину, поддерживающую его. А затем началось...

Содержание

Acer Aspire One A110

Цена: 11 690 руб. (Яндекс Маркет)
Сайт: www.acer.ru

Рекомендуется: Ищущим дешевый ноутбук, только для Интернета
Противопоказан: Для игр, ресурсоемких задач

За исключением серии Eee, нетбуки Aspire One от Acer – самые известные на рынке. На данный момент доступны две модели линейки 110: одна использует флэш-память (SSD), а на другой стоит традиционный жесткий диск. Здесь мы рассмотрели модель с 8-ГБ SSD, которая уже опустилась до весьма заманчивой стоимости – поискав в сети, вы можете найти ее за сущие копейки.

По части косметики, Acer хорошо поработала над формой Aspire One. Крышка и корпус плотно соприкасаются, и при открытии машины вас встретит выпуклая колобашка под экраном с индикаторами статуса и батареи. Между шарнирами, скрепляющими экран и корпус, просматривается зазор, наводя иллюзию их слабости. Однако в целом Aspire One очень прочен, только экран слегка прогибается.

А экран глянцевый. Кое для кого это фактор выбора, но есть и пользователи, люто ненавидящие такие экраны. Если сесть под неправильным углом к свету, экран бликует, затрудняя просмотр. Дома или в офисе это несложно поправить, а вот на улице не очень-то избавишься от большого небесного фонаря. С другой стороны, цвета сочные и яркие, и это отличный дисплей при такой цене.



› **Модель с жестким диском немного толще и намного дороже протестированной SSD-версии.**

Пиршество портов

Слева находятся слот для SD-карты, порты для USB и Ethernet, выход VGA, а справа – добавочная пара USB-портов, еще один слот для SD-карты, гнездо для кенсингтонского замка и порты для наушников/микрофона. Клавиатура отличная – клавиши правильного размера, кнопка Enter широкая, и общее впечатление хорошее. Клавиши курсора маловаты, но зато есть отдельные кнопки Page Up/Down. Сенсорная площадка нас не воодушевила: уж очень мала, и для экономии места дизайнеры Acer расположили клавиши по сторонам. Для работы с площадкой вам придется заново тренировать моторику; мы бы предпочли обычный расклад.

› **Linpus Linux, ОС по умолчанию, прячет от пользователя часть функций.**

Программно, Aspire One укомплектован Linpus Linux Lite – это дистрибутив на базе Fedora, снабженный всем, что полагается: *Firefox*, *OpenOffice.org* и т. д. Приличный дистрибутив, с достойным временем загрузки (22 секунды), однако немного тормозит, да и устарел; многие пользователи меняют его на «полномерные» варианты. Хорошо работает Ubuntu 9.04 Netbook Remix, где Wi-Fi, web-камера и звук оживают сразу после установки.

> «Acer хорошо поработала над формой.»

Еще одно разочарование – скорость SSD. Иногда он еле ползает, в особенности при записи данных. Linux (как и любая современная ОС) копит операции записи файлов для выполнения пакетом, а на Aspire One это проявляется в виде неприятных запинок. Простое переключение между категориями в Ubuntu Netbook однажды заняло 11 секунд, так как в это время SSD завершал очередную работу.

Вообще говоря, это не грозная проблема, если ваша стихия – в основном онлайн: машина превосходна для web-серфинга и легко справляется с YouTube. Частенько включается кулер, а колонки крошечные; ну да при такой цене роскоши нечего и ждать. Если вы планируете работать или играть в игры в пути, поведение SSD будет вас раздражать, а модель с жестким диском стоит заметно дороже. Тогда вам лучше взглянуть на другие тестированные нами модели, например, Eee PC.

LINUX FORMAT Вердикт

Производительность	5/10
Удобство	7/10
Качество сборки	8/10

» Приятная цена и удобство для мобильного Интернета, но медленный SSD-накопитель ужасно бесит.

Рейтинг 7/10

Asus Eee PC 1000



› Прото-нетбук все еще силен, и теперь поставляется в нескольких разновидностях.

Цена: 14 300 руб (Яндекс Маркет)
Сайт: ru.asus.com

Рекомендуется: Ищущим практичности, близкой к ноутбуку
Противопоказаны: Фанатам портативности

Велик, в обоих смыслах этого слова. Не только потому, что Asus – самая известная марка в мире нетбуков, с огромным диапазоном моделей, но также потому, что Eee 1000 стал самой большой машиной в нашем тесте. По размерам он недалек от обычного ноутбука, и довольно тяжел, почти 1,3 кг. То есть основная мысль здесь такая: если вы требуете от нетбука максимум портативности, то Eee 1000 не для вас – сразу переходите к обзорам моделей Toshiba и Dell Mini.

«Цвета живые, а края изображений очень четкие.»

Благодаря массивности, Eee убедительно-прочен. Пластмасса не трещит при надавливании, экран твердый и почти не гнется, а клавиатура не стучит. Первая модель Asus 701 могла запросто выдержать несколько ударов, и мы рады видеть, что компания не изменила солидному дизайну. Часть веса обусловлена мощной стандартной 6-элементной батареей, которая заодно повышает толщину модели – Asus заявляет о времени автономной работы в семь часов, а мы добавим, что это недалеко от истины, особенно если вы не сильно прессуете CPU.

Ненавистники глянцевых экранов, ликуйте: экран у Eee PC 1000 матовый. Тем не менее, цвета живые, края изображений четкие – это один из лучших экранов в нашем тесте. Встроена 1,3-МПикс web-камера, а вверху клавиатуры есть кнопки для переключения видеорежимов (то есть на отдельный дисплей), запуска *Skype* и изменения уровня производительности. Восхищает клавиатура – огромная кнопка Enter и полномерные курсорные клавиши. Единственный недостаток – маленькая и скрипучая правая клавиша Shift.

С такими размерами машины в Asus смогли запихнуть сенсорную площадку сносного размера, притом с большими кнопками. В дистрибутиве Xandros Linux, стоящем по умолчанию, она ведет себя довольно странно; указатель курсора «плавает», смещаясь на несколько пикселей уже после того, как вы отпустили палец. У наших коллег из *PC Format* было то же самое, так что это не дефект экземпляра, взятого для обзора.

Обычные подозреваемые

Слева находится кенсингтонский замок, разъем Ethernet, воздухозаборник, порт USB и гнезда для наушников/микрофона, а справа – вход для шнура питания, слот для SD/MMC карты, выход VGA и еще два USB-порта. Как уже упоминалось, Eee поставляется с Xandros Linux, с четкими и массивными значками на рабочем столе. Это весьма дружественный дистрибутив для новичков, однако набор приложений жутко устарел (*Firefox 2*?!), а менеджер обновлений презабавно слетал, когда мы пытались проверить, нет ли чего новенького.

Как и ожидалось, Ubuntu запустился прекрасно, хоть и не распознал наш модуль Bluetooth. 40-ГБ SSD-носитель на нашей машине был разбит на два диска, один на 8 ГБ, другой на 32 ГБ. На 8-ГБ мы поставили Ubuntu для наших испытаний. При таких размерах Eee мы ожидали, что места хватит для хорошей вентиляции CPU, но увы! Во время тестов кулер включался на каждом шагу, причем с изрядным шумом. Немного досадно, но в остальном было неплохо.



Linux Format Вердикт

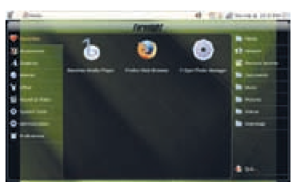
Производительность	8/10
Удобство	9/10
Качество сборки	9/10

» Большой и тяжелый, хотя цена хорошая; емкая батарея, а по удобству близко к ноутбуку.

Рейтинг 9/10

Дистрибутивы для нетбуков

Ubuntu Netbook Remix – выдающийся дистрибутив для нетбуков, но он не одинок. В последних релизах Mandriva принялегла на совместимость с нетбуками, а Debian сделал заначку на http://wiki.debian.org/DebianEeePC. Есть и другие:



› Интерфейс Foresight Linux Mobile такой же, как у UNR.

» **Easy Peasy** Ответвление Ubuntu с проприетарными дополнениями, а именно Flash и *Skype*. www.geteasypeasy.com

» **CrunchEee** Другая производная от Ubuntu, для более опытных пользователей. www.crunchbanglinux.org

» **Foresight Linux Mobile** Использует уникальную систему управления пакетами *Conary*. www.foresightlinux.org/mobile.html

» **Pupeee** Легковесный дистрибутив, хорошо подходящий для слабосильных устройств, типа Asus Eee 701. www.puppylinux.org/wikka/Pupeee

Dell Mini 9

Цена: $249 (Google Product Search)
Сайт: www.dell.com

Рекомендуется: Любителям портативности и тишины
Противопоказан: Тем, кто не желает идти поперек привычной моторики

Бывают ли по-настоящему тихие нетбуки? Вот у Mini 9 нет жесткого диска, а охлаждается он полностью пассивно. Достижение серьезное: ведь у него такой же CPU, как у Eee 1000, а тот – зверюга не из молчаливых. Но все имеет оборотную сторону: низ машины сильно греется при нагрузке на CPU.

Mini 9 не украшен всякими металлическими ободками, как Eee или Aspire One. Поверхность крышки ласкает подушечки пальцев, но экран – глянцевый, и вдобавок отражает, что может печалить при работе на улице. Слева находятся гнезда для кенсингтонского замка и шнура питания, два порта USB и слот для SD/MMC карты; справа – разъемы наушников/микрофона, еще один USB-порт, на диво маленький воздухозаборник, выход VGA и Ethernet.

Dell пошла здесь на несколько смелых решений; самое заметное из них – устранение функциональных клавиш с их привычного места вверху клавиатуры. Вместо этого нужно нажимать Fn+A к клавише точка с запятой на клавиатуре, что требует некоторой переподготовки сознания.

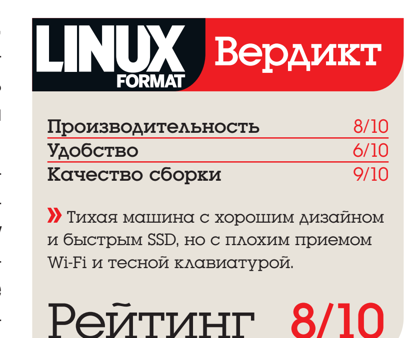
Другая причуда – сдвиг правой клавиши Shift на две кнопки влево, сразу за обратным штрихом (`) и стрелкой вверх. Но в остальном клавиатура спокойная и крепкая; клавиши того же размера, что и у Toshiba NB100, но чуть повыше и от этого намного удобнее. Сенсорная площадка, сделанная впритык к клавиатуре, тоже неплоха.



› Mini 9 очень портативен, правда, за счет тесной клавиатуры.

Dell поставляет Mini 9 с Ubuntu 8.04 LTS, и Netbook Remix 9.04 работает на этой машине с блеском, а быстрая SSD-память выводит его на второе место по скорости загрузки.

Время работы батареи тоже достойное; а по приему Wi-Fi тест дал второе место с конца, докладывая только 41 % силу сигнала, в то время как другие показывали до 70 %. Но это неплохое приобретение для цены $249, хотя за модель с web-камерой придется чуть-чуть доплатить.



LG X110

Цена: 13 799 руб (Яндекс Маркет) **Только Windows!**
Сайт: http://ru.lge.com

Рекомендуется: Опытным пользователям, при двойной загрузке с Windows
Противопоказан: Тем, кому жалко денег на Microsoft

Многие машины этого теста предпочли убогонькие черные и синие тона для своих корпусов и клавиатур, но в этом LG смешиваются мягкие оттенки белого и серебристого. Крышка очень недооценена: она почти вся белая, кроме маленького логотипа LG наверху.

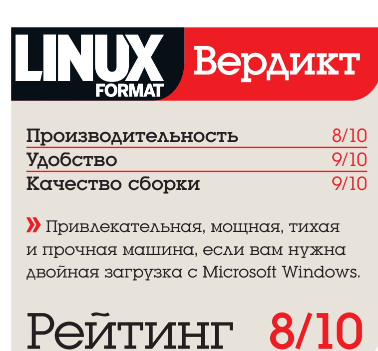
По части оборудования, сразу видно, что X110 – тот же MSI Wind (мы рассмотрим его ниже), с таким же расположением портов и таким же громадным воздухозаборником слева. Однако здесь позаботились о внешности: в Wind разъемы для наушников и микрофона отмечены привычными зеленым и красным ободками, а у X110 они сохраняют цвет корпуса. Вдобавок, порт VGA в X110 черный, а не синий, что придает модели профессиональный вид.

Главная разница X110 и Wind – в качестве клавиатуры и сборки. В X110 они супер: тишина, все клавиши на привычных местах… хотя курсорные все-таки не полномерные. Шарниры в X110 немного слабее, чем, скажем, в Eee или IdeaPad, но все остальное сделано очень хорошо, и выбор материалов лучше, чем в Wind. Нашему обзору подвергся экземпляр с 3-элементной батареей, которая показала плохие характеристики (см. сводные результаты) – рекомендуем потратить лишние пару тысяч рублей и взять версию с 6-элементной батареей.



› Мы бы сменили батарею, а в остальном LG X110 работает хорошо.

При всей схожести в оборудовании, на X110 стоит другой беспроводной чип и нет Bluetooth. Однако мы не станем засиживаться на этой машине – просто потому, что она поставляется только с Windows. Если вы регулярно пользуетесь ОС от Microsoft и хотите иметь нетбук с двойной загрузкой, он прекрасен – хорошо спроектирован и прекрасно работает. Если вам нечего делать с Windows, но по душе характеристики машины, обратитесь к похожему Wind.



Elonex ONEt



› **ONEt выжимает все соки из 2-элементной батареи.**

Цена: £119 (Elonex)
Сайт: www.elonex.com

Рекомендуется: Детям и хакерам, ненавидящим x86
Противопоказан: Всем остальным

Даже если вы раньше никогда не слышали о Elonex ONEt, вы, наверно, видели его в других ипостасях. Видите ли, в Китае есть огромный завод, штампующий нетбуки с вариациями дизайна, но с одинаковой начинкой: 400-Мгц CPU, 128-МБ ОЗУ и 1 или 2 ГБ пространства на флэш-носителе. В Maplin они означены как Minibook, а в других местах – как Skytone Alpha 400.

Elonex предлагает ONEt в трех цветах – черном, зеленом и розовом – и едва взяв его в руки, вы понимаете, что он прочный. Твердая основа и толстая пластмасса создают ощущение стойкости ONEt. Это самый крошечный нетбук в нашем тесте. Он чуть уже, чем Toshiba, и это приводит к проблемам с клавиатурой (особенно с крошечными курсорными клавишами) и с наноскопической сенсорной площадкой. Для экономии места ее кнопки разнесены по сторонам, а-ля Acer Aspire One.

> «Едва взяв его в руки, вы видите, что он прочный.»

Справа имеется пара USB-портов, а слева – слот для SD-карты, разъем для наушников и порт для микрофона. Сзади – другой USB-порт, разъем Ethernet и гнездо питания; неплохой ассортимент при малой цене.

Так что экипировка приличная; а как там с приложениями? ONEt уникален в нашем обзоре тем, что его CPU не является x86-совместимым. На всех других нетбуках теоретически можно запустить ОС от Windows 3.1 до Fedora 11, а ONEt и его собратья используют CPU на основе архитектуры Xburst. Это вариант MIPS, семейства процессоров, широко применяемых SGI в утитанных рабочих станциях Indy и Octane. Таким образом, типовые дистрибутивы Linux и операционные системы для ПК с ONEt работать не будут, что оттолкнет часть покупателей. Однако куда больше проблем доставляет мощность процессора. Чипу Xburst элементарно не хватает силенок на нетривиальные интернет-задачи: он мучительно заикается на сайтах типа BBC News, а с тяжелым JavaScript (как в Google Mail) вообще виснет.

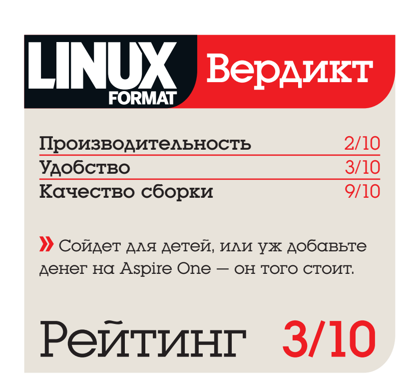
Хромые приложения

Приложения не спасают ситуацию. На ONEt ставится сделанный под заказ дистрибутив CELinux. Он включает *Firefox 2* и, следовательно, медленный JavaScript'овский движок, и из-за ядра MIPS здесь не применим официальный Flash. Интерфейс со вкладками дает доступ к *AbiWord*, *Gnumeric*, четырем играм (включая *Pengupop*) и файловому менеджеру с блокировкой, которая запрещает вам соваться куда бы то ни было вне домашнего каталога.

Есть еще и медиа-плейер, при запуске которого любезно предъявляются максимально допустимые параметры видео (разрешение 350×288 при 128 kbps). Управление питанием отсутствует почти напрочь – есть показатель состояния «ботореи», а подсветка экрана не выключается, когда вы закрываете крышку.

На www.littlelinuxlaptop.com есть несколько дополнений, а также дистрибутив 3MX, созданный сообществом – он весьма неплох для технарей, однако для просмотра web все равно придется запускать *Dillo*. Тем не менее, 3MX – это правильный Linux, и он дает вам чувство контроля.

Итого, ONEt совершенно непригоден для категории «мобильного образа жизни»: он реально пасует при обзоре Интернета. Если Elonex снизит цену долларов до ста и добавит образовательные программы типа *GCompris* и *KGeography*, он подойдет для детей. Однако он слишком маломощен и неэффективен по цене, если только вам не загорелось побаловаться с портативным устройством на базе MIPS.



LINUX FORMAT Вердикт

Производительность	2/10
Удобство	3/10
Качество сборки	9/10

» Сойдет для детей, или уж добавьте денег на Aspire One – он того стоит.

Рейтинг 3/10

Доля MS на рынке

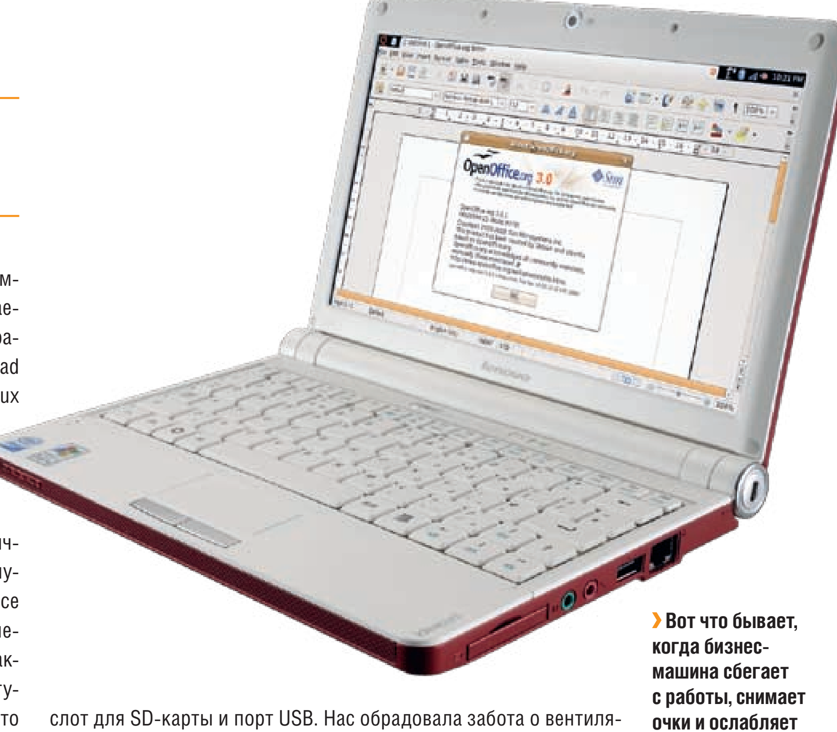
В начале апреля Microsoft гордо провозгласила в своем Windows Blog, что захватила 96% рынка ОС для нетбуков. (http://tinyurl.com/msnetbookshare). Нас, фанатов Linux, это слегка обеспокоило – Canonical опубликовала достойный ответ (http://blog.canonical.com/?p=151), но несомненно, что Linux малость оттеснен в некоторых частях рынка, как видно по ряду нетбуков только для Windows в нашем тесте.

Один отважный участник команды **LXF** пару месяцев назад зашел в PC World и услыхал, как продавщица говорила супружеской паре, пришедшей за нетбуком: «Это не для вас, потому что это Linux, и в нем нельзя запускать ваши программы». Возможно, они сразу и просили нетбук с Windows, и такой совет бил в точку. Однако нас удивляет, почему магазины и центры продаж по всему миру даже не дают Linux настоящего шанса – ну да, конечно, продавая нетбуки с Windows, можно заодно всучить «пакеты безопасности» и все прочие дополнения…

Lenovo IdeaPad S10e

Цена: 15 000 руб (Яндекс Маркет)
Сайт: www.lenovo.com/ru/ru

Рекомендуется: Опытным пользователям, которым нужен слот ExpressCard
Противопоказан: Работающим в тихой обстановке, из-за ультрагремучих кнопок сенсорной площадки!



› Вот что бывает, когда бизнес-машина сбегает с работы, снимает очки и ослабляет галстук. Наверно.

Название Lenovo не часто увидишь на домашних компьютерах, однако в бизнес-кругах компания уважаема за свою крепкую линейку ноутбуков ThinkPad (ранее выпускаемых IBM). Следовательно, на ее нетбуках IdeaPad вы не встретите ни Ubuntu, ни Mandriva – только SUSE Linux Enterprise Desktop. Lenovo прислала наш тестовый экземпляр с Windows XP, но, к счастью, Splashtop затеял показать миру, как силен Linux, и мы сразу установили Ubuntu Netbook Remix.

С точки зрения внешности, IdeaPad напоминает обычный ноутбук для бизнеса, на который глядят через перевернутый бинокль. Нельзя сказать, что он плохо смотрится – вовсе нет. Но у него нет плавных изгибов, как у прочих нетбуков нашего теста, и выглядит он весьма сурово. И мы не в восторге от практичности. Правда, клавиши курсора полноменрные, но клавиатура изрядно дребезжит, а правая кнопка Shift того же размера, что и остальные – работать неудобно (в особенности после огромных клавиш на Aspire One и MSI Wind).

Не туда

Вдобавок клавиша Fn находится на клавиатуре слева внизу, там, где обычно ожидается клавиша Ctrl. Не так уж трудно было бы поменять местами клавиши Ctrl и Fn, чтобы раскладка стала более привычной, как сделано во многих других моделях теста. Что касается сенсорной площадки, она мала. Реально мала. Кнопки твердые, но и гремучие. Эта не катастрофа для практичности, но малость порочит во всем остальном идеально собранную машину, хотя ее глянцевый экран чуть прогибается.

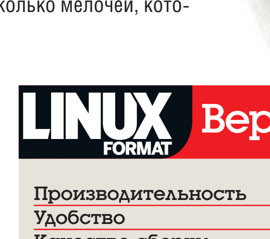
IdeaPad – единственный нетбук в нашем тесте, имеющий слот ExpressCard. Он находится справа, наряду с разъемами для наушников/микрофона, порта USB, Ethernet и гнездом для кенсинг тонского замка (встроено в шарнир). Левая сторона содержит убедительных размеров воздухозаборник, а также выход VGA, слот для SD-карты и порт USB. Нас обрадовала забота о вентиляции – при большой загрузке CPU машина настолько раскаляется, что к днищу не притронуться.

Трубчатый выступ

Lenovo нашего теста поставлялся с 6-элементной батареей, образующей трубчатый выступ вдоль задней части машины; представьте себе пару склеенных скотчем упаковок Mentos, и вы поймете, о чем речь. Результаты наших испытаний были впечатляющими: при интенсивном тесте батарея работала 4 часа 16 минут. При неинтенсивном использовании ее хватает на рабочий день, и зарядное устройство можно с собой не таскать.

> «В нашем тесте батарея протянула 4 часа 16 минут.»

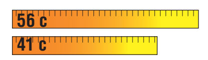
В плане приложений, Ubuntu Netbook Remix работает прекрасно, а 160-ГБ жесткий диск производит операции быстро, если сравнивать с виденными нами вялыми SSD. Довольно хороши басы колонок, расположенных на устройстве спереди. В общем, Lenovo – неплохая машина, однако есть несколько мелочей, которые способны испортить вам жизнь.



Приводы за большие деньги

За SSD (Solid State Drives – они же твердотельные накопители и флэш-диски) определенно стоит будущее, и мы ожидаем, что они за следующие несколько лет постепенно вытеснят магнитные винчестеры. Однако емкие и быстрые диски сейчас довольно дороги. Мы остановились на OCZ Vertex 120GB SSD, который продается примерно за 20 000 рублей, и поставили его на MSI Wind. Вот результаты испытаний:

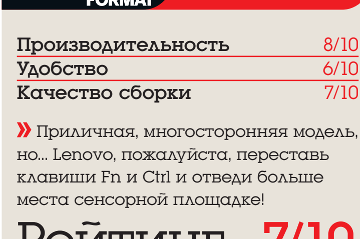
MSI Wind (обычный)
MSI Wind (с приводом OCZ)



Разница невелика. Запись 100-МБ файла заняла две секунды, что, несомненно, неплохо для SSD, но не особо опережает жесткие диски. Пока эти устройства не упадут в цене, мы не станем рекомендовать их для нетбуков: это перебор.



› Вот самый яркий индикатор активности жесткого диска в мире нетбуков. Иногда он отвлекает.



LINUX FORMAT **Вердикт**

Производительность	8/10
Удобство	6/10
Качество сборки	7/10

» Приличная, многосторонняя модель, но... Lenovo, пожалуйста, переставь клавиши Fn и Ctrl и отведи больше места сенсорной площадке!

Рейтинг 7/10

MSI Wind U100

Цена: 14 500 рублей (Яндекс Маркет)
Сайт: http://ru.msi.com

Рекомендуется: Опытным пользователям
Противопоказан: Для грубого обращения

› В наших испытаниях MSI Wind U100 показывал хорошие результаты.

Название MSI (Micro-Star International) тоже нечасто встретишь в доме: тайваньская фирма в основном перепродает свои компьютеры другим поставщикам, а те добавляют свои логотипы и названия. Мы уже видели такое на примере LG X110, однако иногда MSI хочется работать в одиночку, и, как следствие, появилась серия нетбуков Wind. (У MSI есть еще и линейка настольных компьютеров под названием Wind.)

> «UNR настроил Bluetooth прямо из коробки.»

По корпусу U100 с ходу видно: он буквально притягивает отпечатки пальцев. Не то что их сложно стереть тряпкой, но на косо падающем свету пятна отчетливо заметны, и ваша электроника выглядит неопрятно. Слева расположены огромный воздухозаборник, гнездо для кенсингтонского замка, разъем для питания и два USB-порта. Справа будет другой порт USB, слот для карты SD/MMC, разъемы для микрофона и наушников, VGA-выход и Ethernet.

Под исключительно крепким экраном расположена тугая, не клацающая клавиатура, которая, подобно Lenovo IdeaPad, имеет клавишу Fn в левом нижнем углу, на месте Ctrl, и если вы используете эту кнопку часто, потребуется время на привыкание. Правый Shift – слева от курсорной клавиши вверх, что нормально для настольных компьютеров, но необычно для нетбуков. В остальном клавиатура отличная. Сенсорная площадка достаточно глубока, ее кнопки почти упираются в край – хотя на фотографии они выглядят как одна кнопка, их на самом деле две. Хорошо бы MSI сделал сенсорную площадку пошире; места хватит.

Вкл питания

MSI снабдила наш Wind 6-элементной батареей (плюс 3 500 рублей), которая выпирает снизу на 1 см и добавляет веса, приближая нетбук по параметрам к Eee 1000. Однако у нас есть замечания по качеству сборки. Оно неплохое – нетбук отнюдь не выглядит способным развалиться у вас в руках. Но пластик трещит при нажатии, и шарниры не так прочны, как хотелось бы. Корпус хороший, и у аккуратных владельцев проблем, наверное, не будет, но нам было спокойнее с главным конкурентом в этом классе, Asus Eee PC 1000.

Распознавание оборудования

Wind – один из немногих нетбуков в нашем обзоре, где есть Bluetooth, и Ubuntu Netbook Remix настроил его прямо из коробки. Правда, web-камера не распозналась. Что касается приема беспроводного сигнала, времени загрузки и общей производительности, здесь Wind играет почти на равных с Asus Eee PC, хотя SSD в Eee медленнее, чем жесткий диск Wind.

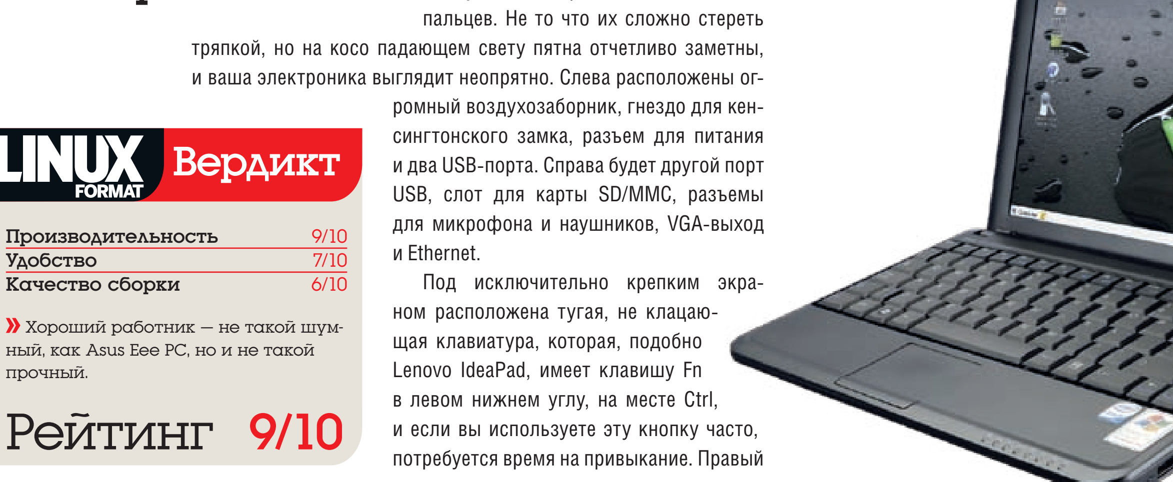
Если вам нужен нетбук побольше, почти ноутбучного размера, и с огромной батареей, два варианта для вас – Wind и Eee. С точки зрения производительности сложно отдать кому-нибудь из них предпочтение, однако у Eee клавиатура лучше, корпус крепче и цена чуть ниже, так что мы порекомендуем его. MSI, впрочем, тоже показал себя неплохо.

LINUX FORMAT Вердикт

Производительность	9/10
Удобство	7/10
Качество сборки	6/10

» Хороший работник – не такой шумный, как Asus Eee PC, но и не такой прочный.

Рейтинг 9/10



Toshiba NB100 11R

Цена: 14 400 руб (Яндекс Маркет)
Сайт: http://www.toshiba-russia.com

Рекомендуется: Бизнесменам, которым нужен максимум портативности
Противопоказаны: Людям с крупными пальцами



› Крошечная клавиатура осложняет использование NB100 11R взрослыми людьми.

Теперь мы угодили в страну лилипутов. Когда в наши руки впервые попал самый первый Eee 701, его клавиатура показалась нам тесноватой, но годной для печати буква за буквой. Затем, по мере роста рынка нетбуков, экраны с диагональю девять дюймов стали нормой, и все поняли, что увеличение длины клавиатуры на какую-то пару сантиметров может стать решающим. А здесь не та ситуация. Клавиатура Toshiba NB100 аналогична Eee 701, хотя сильнее стучит и клацает, и если у вас большие руки, машинка покажется вам неудобной по сравнению с другими.

Но не считайте это ошибкой Toshiba: компания борется за портативность. По малой ширине это второй нетбук (после ONEt) из нашего теста, с отрывом от Dell Mini 9 на целый сантиметр. Он чуть повыше, чем Dell, хотя в основном из-за необычной батареи, которая выпирает с тыла (об этом чуть позже). Внешне NB100 очень похож на IdeaPad – он тоже выглядит как уменьшенный ноутбук. Однако экран у него глянцевый, который хорошо выглядит под определенными углами, а под другими ужасно бликует.

В плане размещения портов Toshiba проявила больше креативности, чем ее соперники. Разъемы под микрофон и наушники находятся спереди машины, слева расположены один USB-порт и гнездо кенсингтонского замка, справа – пара дополнительных портов USB и слот для SD/MMC, а Ethernet, питание и VGA – сзади. Качество сборки первоклассное, если не считать шумную клавиатуру: все твердое и прочное.

Теперь про батарею...

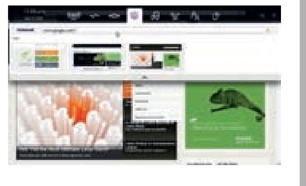
Батарея торчит из корпуса примерно на сантиметр. Не такая уж беда, верно? В других нетбуках тоже есть 6-элементные батареи, и они тоже выпирают! Да только у NB100 батарея 4-элементная, и выдается она, потому что узкая. Например, у Aspire One батарея тянется примерно на 90 % ширины машины; в NB100 параметр близок к 60 %. Дизайнеры Toshiba действительно хорошо поработали, чтобы втиснуть побольше в малый объем, но мы полагаем, что широкая батарея, вписывающая под заднюю часть, лучше, чем более узкая, которая создает выступ.

> «Toshiba хорошо поработала, чтобы втиснуть побольше.»

Переходим к производительности. С ней не без проблем. Скорость загрузки не самая лучшая, а качество приема беспроводного сигнала варьируется около критического уровня, даже когда нетбук находится в покое, что приводит к слабым результатам по беспроводной части в таблице на с. 24. Работа в Ubuntu Netbook Remix показалась сбивчивой – видеофильмы спотыкались, и задержки при открытии файлов превысили ожидания. Это не делает машину непригодной, однако может раздражать.

С учетом всего этого, Toshiba NB100 11R будет трудно выдержать конкуренцию в тестовой группе. Он очень компактен и имеет солидный, деловитый фасад, так что если вам нужно прихватить что-нибудь на деловую встречу и сделать кое-какую офисную работу, это прочная и тихая машинка. Но большинству пользователей мы рекомендуем пожертвовать портативностью – всего дюймом ширины – ради удобства работы.

Moblin и SUSE



Moblin, мобильная ОС от Intel на базе Linux OS (см. *Что за штука* в **LXF118** и материал в **LXF119**), получила добрый толчок в плечо, благодаря помощи команды Novell SUSE. Пока мы писали эту статью, Novell готовилась продемонстрировать свое «внедрение Moblin в кодовую базу OpenSUSE», как сказал нам Хольгер Дюрофф [Holger Dyroff] из подразделения Business Development.

Теперь SUSE поставляется с десятком нетбуков от MSI и HP. Улучшив версию Moblin, Novell надеется захватить больше рынка, в особенности деловой сегмент. «Предприятия могут рассматривать нетбуки как замену тонких клиентов», сказал Дюрофф, а компания имеет «большой интерес сделать на этом бизнес».

Станет ли он очередным соперником для Ubuntu Netbook Remix? Пока вы читаете эти строки, Novell должна выложить свою работу в открытый доступ на http://en.opensuse.org/Moblin – зайдите и посмотрите.



LINUX FORMAT Вердикт

Производительность	5/10
Удобство	5/10
Качество сборки	8/10

» Подходит для бизнеса, однако производительность не очень и клавиатура маловата.

Рейтинг 6/10

Кто лучше? Кто лучший?



Почти все наши нетбуки основаны на одинаковых чипах Intel, и тем не менее разнообразны по размеру, весу, качеству сборки, скорости диска и цены. Если не учитывать слабый CPU у Elonex ONEt и летаргический SSD у Aspire One, они не сильно отличаются по производительности; все, кроме ONEt, пригодны для обзора Интернета, YouTube и офисной работы, и даже позволяют программировать на ходу, если только вы не проектируете чрезмерно ресурсоемкое 3D-приложение.

В каждом обзоре мы рассматривали машину с точки зрения определенного типа пользователя для каждой модели, и вы, наверное, уже поняли, что подойдет лично вам. Если вас интересует исключительно финансовая сторона, вот наши рекомендации:

» **Дешево** Aspire One. Без вариантов. Даже если вы увидите ONEt (или другой нетбук с той же начинкой) еще дешевле, пройдите мимо. Экономия денег не восполнят страданий от медленного web-серфинга, многие сайты останутся вне досягаемости.

» **Средне** Если решающую роль играет портативность, выбирайте Dell Mini 9. Придется помучиться с клавиатурой, однако это хорошо разработанная и сконструированная машина. Если стерпите что-нибудь более громоздкое и шумное, возьмите Eee 1000 – хороший, универсальный нетбук.

» **Дороже** Нам очень понравился LG X110; жаль, что он доступен только с Windows. Если вы соблазнились, следите за онлайн-новостями: вдруг LG расширит возможности и начнет предлагать Linux-версию, тут уж его нельзя не купить. Иначе довольствуйтесь более слабым, но все равно неплохим MSI Wind.

Конечно, если у вас уже есть какой-то из рассмотренных здесь нетбуков, мы будем рады услышать ваше мнение! Зайдите на наши форумы на www.linuxforum.ru и поделитесь опытом – какую конкретную модель вы полюбили? Вам нравится радуга цветов в ONEt? Сумели заставить работать в Mini 9 функциональные клавиши? Дайте нам знать! LXF

Время загрузки

Измерялось от нажатия кнопки питания до появления рабочего стола и прекращения активности диска. Все проверялось на UNR 9.04, кроме Elonex ONEt.

Acer Aspire One	78 с
Asus Eee 1000	43 с
Dell Mini 9	42 с
Elonex ONEt	33 с
Lenovo IdeaPad	54 с
LG X110	56 с
MSI Wind U100	56 с
Toshiba NB100	63 с

Срок работы от батареи

Мерился при отключенном энергосбережении, полной яркости экрана, подключении к Wi-Fi и зацикленном воспроизведении видео с локального диска.

Acer Aspire One	1 час 43 мин.
Asus Eee 1000	3 часа 42 мин.
Dell Mini 9	2 часа 52 мин.
Elonex ONEt	3 часа 7 мин.
Lenovo IdeaPad	4 часа 16 мин.
LG X110	1 час 32 мин.
MSI Wind U100	3 часа 21 мин.
Toshiba NB100	3 часа 5 мин.

Скорость записи носителя

Определялась путем локальной записи 100-МБ файла и последующим выполнением **sync** для сброса изменений на диск. Усреднена по трем опытам.

Acer Aspire One	27 с
Asus Eee 1000	8 с
Dell Mini 9	6 с
Elonex ONEt	73 с
Lenovo IdeaPad	4 с
LG X110	3 с
MSI Wind U100	3 с
Toshiba NB100	3 с

Беспроводной прием

Показания качества сигнала из состояния апплета *NetworkManager* (на Elonex ONEt – команда **iwlist**). Точка доступа находилась в шести метрах, за кирпичной стеной.

Acer Aspire One	48
Asus Eee 1000	72
Dell Mini 9	40
Elonex ONEt	86
Lenovo IdeaPad	70
LG X110	68
MSI Wind U100	74
Toshiba NB100	38







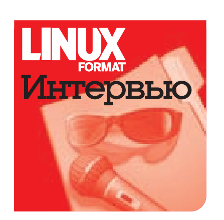



Название	Acer Aspire One A110	Asus Eee PC 1000	Dell Mini 9	Elonex ONEt	Lenovo IdeaPad S10e	LG X110	MSI Wind U100	Toshiba NB100 11R
Цена	11 690 руб.	14 300 руб.	$249	£119	15 000 руб.	13 799 руб.	14 500 руб.	14 400 руб.
Размер, мм	248×170×32	266×190×38	233×168×33	213×143×32	250×184×34	261×181×35	258×182×40	224×190×37
Масса, г*	910	1290	1000	610	1050	1150	1225	1020
Дисплей	8,9", 1024×600	10", 1024×600	9", 1024×600	7", 800×480	10,1", 1024×576	10", 1024×600	10,2", 1024×600	8,9", 1024×600
Видео	Intel 945GME	Intel 945GME	Intel 945GME	JzSOC чип	Intel 945GME	Intel 945GME	Intel 945GME	Intel 945GME
CPU	1,6 ГГц Intel Atom N270	1,6 ГГц Intel Atom N270	1,6 ГГц Intel Atom N270	400 МГц XBurst	1,6 ГГц Intel Atom N270	1,6 ГГц Intel Atom N270	1,6 ГГц Intel Atom N270	1,6 ГГц Intel Atom N270
ОЗУ	512 МБ	1 ГБ	1 ГБ	128 МБ	1 ГБ	1 ГБ	1 ГБ	512 МБ
Диск, ГБ	8, SSD	40, SSD	8, SSD	2, SSD	160, HD	160, HD	80, HD	80, HD
Батарея	3 элемента, 2200 мАч	6 элементов, 6600 мАч	4 элемента, 2200 мАч	2 элемента, 2100 мАч	6 элементов, 4360 мАч	3 элемента, 2200 мАч	6 элементов, 5200 мАч	4 элемента, 3800 мАч
Порты	3×USB, 1×SD, 1×SD/MMC, Ethernet, VGA	3×USB, 1×SD/MMC, Ethernet, VGA	3×USB, SD, Ethernet, VGA	3×USB, SD, Ethernet	2×USB, SD/MMC, VGA, Ethernet, ExpressCard/34	3×USB, SD/MMC, Ethernet, VGA	3×USB, SD/MMC, Ethernet, VGA	3×USB, SD/MMC, VGA, Ethernet
Web-камера	0,3 MP	1,3 MP	1,3 MP	Нет	1,3 MP	1,3 MP	1,3 MP	0,3 MP
Беспроводная связь	Atheros AR242x	RaLink RT2860	Broadcom BCM4312	ZyDAS ZD1211B	Broadcom BCM4312	Realtek RTL8187SE	RaLink RT2860	Atheros AR242x
Bluetooth	Нет	Есть, но не распознавалась Ubuntu	Нет (приобретается отдельно)	Нет	Нет**	Нет	Есть	Нет
ОС	Linpus Linux Lite	Xandros Linux	Ubuntu 8.04	Дистрибутив, основанный на CELinux	SUSE Linux Enterprise Desktop	Windows	SUSE Linux Enterprise Desktop	Ubuntu 8.04

* С батареей ** Только на Windows-версии

MathGL: Данные тоже нуждаются в представлении

Данные могут быть прекрасны не только с точки зрения исследователя. Для этого их следует хорошенько подготовить, и *MathGL* поможет здесь даже самому взыскательному эстету. Этот довольно очевидный факт теперь признали и во Франции. **Евгений Балдин** беседует с **Алексеем Балакиным**, победителем Les Trophées du Libre 2009.



Первая идея о том, что следует написать нечто вроде *MathGL*, посетила меня в районе 2003 года, когда мне впервые пришлось столкнуться с крупными (трехмерными) массивами данных и я осознал все проблемы, связанные с их визуализацией. На тот момент единственным пригодным инструментом являлся *MATLAB*... но и у него была, да и до сих пор остается куча недостатков: нестабильная работа с большими массивами, высокие накладные расходы при рисовании графиков (памяти надо много больше, чем занимает исходный массив) и, естественно, высокая цена (хотя тогда для меня это было не критично, так как все ПО было закуплено работодателем).

Еще одна причина, подвигнувшая меня на написание *MathGL*, была опять же связана с расчетами. Получающиеся в результате них многогигабайтные файлы было очень неудобно передавать по сети (с кластера на локальную машину) только для того, чтобы отрисовать несколько картинок. Из-за этого неудобства возникало естественное желание строить графики и считать данные в одном месте.

Собственно, ориентация на большие массивы данных (в том числе, и высокой размерности), а также возможность построения картинки в любом окружении и легли в основу *MathGL*.



» **КТО** Алексей Антонович Балакин, кандидат физ.-мат. наук, старший научный сотрудник Института прикладной физики Российской академии наук (ИПФ РАН).

» **ОТКУДА** Родился, живет и работает преимущественно в Нижнем Новгороде. Закончил Нижегородский государственный университет.

» **ПРОЕКТ** MathGL (mathgl.sf.net).

» **РЕЗУЛЬТАТ** Первое место на Les Trophées du Libre 2009 в категории Sciences.

После того как предметные требования к программе были сформулированы, настало «дело техники». В 2005–2006 годах была написана первая версия для личного использования. Кстати, именно она применялась при подготовке важнейшего результата РАН в 2006 году («Исследование структурных особенностей динамики самофокусировки сверхкоротких импульсов с шириной спектра порядка центральной частоты»). В 2007 *MathGL* слегка расширилась и стала публичной. С тех пор идет постоянное улучшение (добавление новых типов графиков, возможностей, и т. д., и т. п.), основанное преимущественно на предложениях пользователей. На данный момент в *MathGL* только базовых типов графиков насчитывается более 50 штук (!).

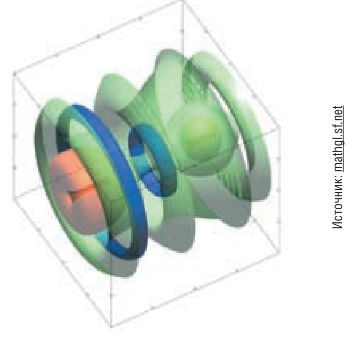
Из того, чем стала сейчас *MathGL*, довольно весомая часть принадлежит и сообществу. Оно не только выдает предложения о новых возможностях и докладывает об имеющихся ошибках, но и вносит непосредственный вклад в сам открытый код. Например, Михаил Видасов добавил возможность для экспорта графики в формат U3D (Universal 3D). При включении U3D-графика в PDF, *Adobe Reader* (начиная с версии 8.1) позволяет мышью вращать, приближать и менять прочие его параметры прямо в файле, что увеличивает интерактивность электронных документов и презентаций.

MathGL изначально задумывалась как кросс-платформенная библиотека, так как мне надо было строить графики непосредственно на сервере/кластере, которые все (по крайней мере, за границей) работают под управлением Linux [и не за границей тоже, – *прим. ред.*]. В то же время схожий функционал был мне необходим и под Windows на локальной машине/ноутбуке.

Что за штука MathGL?

MathGL – это библиотека, предназначенная для построения широкого спектра высококачественных научных графиков (кривых, поверхностей, поверхностей уровня и так далее). Библиотека является кросс-платформенной. *MathGL* предоставляет возможность экспорта графиков в растровые (PNG, JPEG) или векторные (EPS, SVG, IDTF) форматы и оптимизирована для работы с большими массивами данных. Сайт проекта: http://mathgl.sourceforge.net/.

«Сферических в вакууме» пользователей может заинтересовать созданная с использованием *MathGL* программа построения графиков и изображений *UDAV*. Ее можно найти по адресу http://udav.sourceforge.net/.



Источник: mathgl.sf.net

› **Задача *MathGL* – строить высококачественные графики.**

С Unix-подобными системами я познакомился на первых курсах университета (1992 год), потом были расчеты на кластерах под Linux. В качестве настольной системы я попробовал его в 2003 году, когда начал задумываться о *MathGL*. Однако окончательный переход под Linux у меня произошел довольно поздно (два года назад) и поспособствовала этому, как ни странно, Microsoft. На очередном ноутбуке стояла Vista, которая была настолько «глючной» (особенно при разработке программ) и прожорливой по ресурсам, что, помучившись с ней месяц, я ее снес и поставил Debian (хотя на тот момент полная поддержка всего оборудования отсутствовала). С тех пор не испытываю никаких проблем с безопасностью, стабильностью и прочим, и только «посмеиваюсь» различным вирусным атакам и мучениям с созданием программ для Windows. Кстати, сдать Vista назад мне так и не удалось – правда, я не сильно и старался.

Путь к славе

Про Les Trophées du Libre узнал из новости на LOR (www.linux.org.ru). Подумал, почему бы не попробовать… С визой проблем не было, так как пару раз в год бываю за границей, и билеты они оплачивают, да и язык прилично знаю. Единственная сложность – это подгадать рабочую поездку и поездку на конкурс.

Финал конкурса проходил в Суассоне [Soissons] – маленьком, но довольно приятном городке в 90 километрах к северо-западу от Парижа. Франция мне понравилась, так как это первая европейская страна (не считая чисто туристических Праги и прочих), на архитектуру которой можно посмотреть с удовольствием. И первая, в которой есть нормальные леса (а не лесопарки; по крайней мере, мне так показалось). К сожалению, график был слишком плотный, и побродить мне не удалось… может быть, в следующий раз, благо победителей приглашают в жюри [*улыбается*].

Французы очень вежливые и далеко не столь «педантичные», как немцы (правила на дороге нарушают, хотя бы и по мелочи). Хорошие люди. А вот по-английски говорят далеко не все, что может создать проблемы, особенно если вы селитесь в не слишком дорогой гостинице или берете такси (лучше написать, куда вам нужно попасть, на бумажке). Но говорящих по-английски (хотя и с сильным акцентом) много, так что на улице не заблудитесь.

Сам конкурс в целом прошел хорошо, хотя были и досадные накладки: все-таки не хватает опыта организации крупных мероприятий. Были довольно интересные проекты. Из того, что особенно понравилось – это игра *Neverball* (neverball.org). Интересно и необычно, рекомендую. Честно говоря, совершенно не понимаю, почему ей не дали первое место [она заняла второе, – *прим. ред.*] в категории Hobbies.

В моей категории (Sciences) у *MathGL* почти не было конкурентов. Один проект относился скорее к мультимедиа, а второй был слишком узкоспециализированным (да и мне кажется, что сходные инструменты есть почти в каждом университете – правда, как правило, закрытые).

Судейство было вполне приличным, хотя многое, естественно, зависело от предварительного впечатления и от презентации (это важно!). Предвзятый человек вполне может увидеть некоторое «перетягивание одеяла» на французские проекты, так как им позволяли говорить чуть подольше, да и рассказывали иногда по-французски. Но если проект качественный и есть опыт докладов, как, например, у меня [*улыбается*], то выиграть можно.

Отдельно хотелось бы отметить, что на конкурсе было, помимо моего, еще четыре проекта, выполненных русскоязычными авторами: *UniConvertor* Игоря Новикова (**LXF111**), *Inquisitor* Михаила Якшина (**LXF119**) и *GSQL* Тараса Халтурина. Очень хорошие проекты, за которые не стыдно. Дерзайте!

О РОЛИ РЕДМОНДА:
«Моему переходу на Linux поспособствовала Microsoft.»



› Игорь Новиков (слева) и Алексей Балакин (справа) с наградами за первые места в своих категориях.

Советы номинантам

Если у вас есть желание поучаствовать и выиграть в конкурсе в будущем, то прежде всего следует тщательно подготовить презентацию. А лучше иметь их две-три: одну – краткую и яркую – для публики, и одну большую – для интересующихся. В «большой» презентации можно представить и историю, и структуру программы, и математическую модель, и прочее. А вот в краткой должны быть только картинки и результаты (плюс слайд, поясняющий, зачем это нужно). Учтите, что в реальности времени будет мало (оно тратится на переключение с компьютера на компьютер, на то, чтобы собраться с мыслями, и прочее), и жюри не очень-то интересно слушать проекты не по профилю. Ваша задача – «зацепить» человека, и лучше это сделать первыми слайдами. А уж все детали – потом, так как время для вопросов – это уже время сверх вашего доклада. Еще одно замечание: не надейтесь показать работу программы жюри, так как времени не будет, да и не оценят. Лучше делать презентацию (можно с анимацией) и хорошо ее проговорить.

Ну и помните про формальные критерии. Это функциональность (возможности проекта должны удовлетворять широкий круг пользователей), стабильность и утилитарность (проектом должно быть удобно пользоваться, и хорошо бы без Segmentation fault), новизна и документация («перепевы» на старую тему не приветствуются, и вообще, надо еще разобраться, как проектом пользоваться).

Насчет документации: она должна быть, и она должна быть на английском! Да, тратится много времени, ну и просто лень. Но вы уверены, что в вашей безусловно замечательной программе можно разобраться с наскока? Есть сомнения? Тогда пишите документацию. И вообще, это сэкономит время в будущем, так как всегда можно послать человека читать ее, если вам лень отвечать на вопросы [*улыбается*]. Наконец, когда пишешь документацию (обычно после создания каркаса), то еще раз проходишь по проекту свежим взглядом и выявляешь «нелогичные» и неудобные места, которые следует поправить.

Ну и, конечно, главный рецепт победы – это делать хорошие проекты. Больше свободных проектов, хороших и разных! LXF

Git: /etc под контролем

Хороший администратор всегда сохранит резервную копию конфигурационного файла, прежде чем внести в него изменения. Оказывается, это можно делать автоматически, поясняет **Евгений Зобнин**.



Наш эксперт

Евгений Зобнин Ежедневно борется с халатностью коллег-администраторов и выработал комплексную методику для принуждения к журналированию действий и объяснения причин изменений в конфигурации.

Любой, кто не брезгует редактированием системных конфигурационных файлов вручную, наверняка сталкивался с ситуацией, когда после очередного подкручивания настроек программа или даже вся система начинали вести себя некорректно. Обычно проблему легко решить, просто вернув конфигурацию в первоначальное состояние, но что если изменения были произведены уже давно и вы не помните, что и где исправили, а странности в поведении системы заметили только сейчас? А если вы новичок и отредактировали целую пачку файлов вслепую, следуя какому-то устаревшему HowTo и не до конца понимая все тонкости настройки Linux? Системным администраторам в этом плане еще сложнее: нужный конфигурационный файл мог быть исправлен другим человеком, который даже и не подумал сообщить о произведенных изменениях. Борьба с ошибками настройки может стать настоящей мукой для неподготовленного пользователя, а для некоторых выливается в полную переустановку операционной системы.

Существует несколько способов борьбы с описанной проблемой, самый эффективный из которых – перевести каталог **/etc**, содержащий основные конфигурационные файлы, под управление системы контроля версий. Да-да, именно той, которую используют программисты для фиксации изменений в коде. Система контроля версий позволит оставлять комментарий для каждого действия, произведенного над каталогом **/etc**, и вести историю всех изменений; она обеспечит возможность быстрого отката любого количества правок; она легка в установке и проста в использовании.

Машина времени для ваших файлов

Система контроля версий (Version Control System, VCS) работает по принципу моментальных снимков. Вы вносите некоторое количество изменений в свои файлы, а затем просто вызываете команду, которая «приказывает» VCS сделать снимок рабочего каталога и поместить его в специальное хранилище, называемое репозиторием. Если в будущем вы поймете, что совершили ошибку, и прошлый вариант был лучше текущего, VCS позволит вернуть файлы к тому состоянию, в котором они находились на момент «фотографирования». Пользуясь ею, вы будете уверены, что непоправимых ошибок не бывает и любой, даже удаленный, файл до сих пор существует в репозитории.

› ***Git*** **легко подсветит изменения, внесенные в конфигурационные файлы вами или кем-то еще.**

Изначально VCS применялась только программистами для совместной работы над проектом, и даже сама идея контроля версий принадлежит программистам. Сегодня же VCS используют многие: от писателей и журналистов, следящих с ее помощью за своими текстами, до системных администраторов, использующих VCS для контроля над конфигурационными файлами.

Приступаем

В качестве VCS мы будем использовать *Git*, созданную самим Линусом Торвальдсом [Linus Torvalds] – мы говорили о ней в **LXF116** и **120**. Она быстра, удобна и, что самое важное, не требует создания выделенного репозитория: он может храниться прямо в рабочем каталоге, в нашем случае – **/etc**. Чтобы установить *Git*, просто найдите его в графическом менеджере пакетов и нажмите кнопку Install [Установить] или воспользуйтесь командой **apt-get install git-core** (Debian/Ubuntu) или **yum install git** (Fedora/Red Hat).

Перевести каталог конфигурационных файлов на использование *Git* очень просто: достаточно создать новый репозиторий и добавить в него все содержимое **/etc** (то есть сделать первый снимок). Первая операция выполняется с помощью двух простых команд:

```
# cd /etc
# git init
```

Чтобы никто, кроме root, не смог заглянуть в наш репозиторий и выудить из него ценную информацию (пользовательские пароли, содержимое файла **/etc/sudoers** и т. д.), лишим всех сторонних пользователей любых прав в отношении хранилища:

```
# chmod og-rwx .git
```

Теперь мы должны добавить в репозиторий конфигурационные файлы, но в каталоге **/etc** всегда имеется несколько файлов, которые генерируются динамически или создаются для временных целей. Это может быть, например, файл **/etc/mtab**, содержащий список всех смонтированных в данный момент файловых систем, **/etc/motd** («сообщение дня»), обновляемый во многих системах в процессе загрузки, или кэш **/etc/blkid.tab**, используемый одноименной командой. В Debian/Ubuntu это еще и все файлы с расширением**.dpkg-new** и**.dpkg-old**. Такие файлы могут создать (и обязательно создадут) большую путаницу между содержимым репозитория и актуальным состоянием каталога **/etc**, поэтому мы добавим их имена в «список игнорирования» *Git*:

```
# cat > .gitignore << EOF
*~
*.dpkg-new
*.dpkg-old
blkid.tab
mtab
motd
ld.so.cache
asound.state
adjtime
EOF
```

Теперь можно зафиксировать содержимое **/etc** в хранилище:

```
# git add .
# git commit -a -m «Первый снимок»
```

Вот и все: конфигурационные файлы теперь находятся в репозитории, так что после редактирования одного из них необходимо будет обновить и его. Делается это с помощью одной-единственной команды, независимо от того, какой конкретно файл был изменен:

```
# git commit -a -m «Описание внесенных изменений»
```

Внеся несколько изменений, вы сможете посмотреть их список с помощью команды **log**:

```
# git log
```

Для отмены внесенных изменений воспользуйтесь командой **git checkout**, которая вернет состояние каталога к указанной точке:

```
# git checkout «@{30 minutes ago}»
```

Вы можете задать также и абсолютное время или конкретный снимок, ключ которого можно выудить из строчки **commit** вывода команды **git log**. Главное, не забудьте после этого синхронизировать состояние каталога с репозиторием с помощью команды **git commit**.

Остальные команды, поддерживаемые *Git*, вам, скорее всего, никогда не понадобятся, ведь все изменения, производимые пользователем над каталогом **/etc**, сводятся к простому редактированию файлов. Другое дело – пакетные менеджеры, которые могут добавить в систему конфигурационный файл устанавливаемой программы или удалить уже существующий. Чтобы избежать описанных выше проблем с несоответствием репозитория реальному содержимому каталога **/etc**, последний придется обновлять каждый раз после установки нового пакета:

```
# apt-get install tuxracer
# cd /etc
# git add .
# git commit -a -m "Изменения внесены пакетом tuxracer"
```

Утомительно и неудобно, не спорю. К счастью, есть способ получше.

Используем etckeeper

Набор скриптов под названием *etckeeper* («хранитель etc») создан с целью избавить пользователей от необходимости обновлять репозиторий **/etc** после каждой установки, удаления или обновления пакетов. Он интегрируется с системами управления пакетами дистрибутивов Debian/Ubuntu, Fedora/Red Hat, Arch Linux и выполняет все необходимые действия автоматически.

Пакет *etckeeper* уже доступен для Debian и Ubuntu, поэтому, чтобы начать использовать его в этих дистрибутивах, достаточно выполнить следующую последовательность действий:

```
# apt-get install etckeeper
# etckeeper init
# cd /etc
# git commit -m "Первый снимок"
```

Теперь вы сможете спокойно работать с каталогом **/etc**. После внесения изменений в конфигурационные файлы по-прежнему придется обновлять репозиторий, но установка и удаление пакетов не потребуют каких-либо дополнительных действий: все изменения будут фиксироваться автоматически.

Если в один прекрасный день вы поймете, что контроль версий конфигурационных файлов вам не нужен, просто перейдите в каталог **/etc** и выполните команду **etckeeper uninit**, и репозиторий *Git* исчезнет с вашего диска.

Снимки по расписанию

Если же необходимость запуска специальных команд после каждого редактирования конфигурационных файлов вас пугает, то вы можете настроить систему на создание ежечасных снимков каталога **/etc**. Сделать это просто – достаточно только написать совсем небольшой скрипт, который будет запускаться демоном *cron*:

```
#!/bin/sh
cd /etc
git add .
git commit -m "Автоматический снимок от: `date`"
```

Поместите его в файл **/etc/cron.hourly/etc-git**, установите бит исполнения (**chown a+x /etc/cron.hourly/etc-git**), и вы будете получать новый снимок системных конфигурационных файлов каждый час.

```
commit 1e73ec1d63d6a8e6c690b8c3a824a9df09816d4e
Author: Charlie <root@localhost.(none)>
Date:   Sat Apr 25 01:10:32 2009 +0600

    Открыл доступ к машине по ftp

:100644 100644 16f9e58... e3b3ae3... M  inetd.conf

commit 13d7448545c6d3899bed43a700fc1f2b6073002a
Author: Charlie <root@localhost.(none)>
Date:   Sat Apr 25 01:07:55 2009 +0600

    Закомментировал DNS-сервер провайдера в resolv.conf

:100644 100644 fa7ce14... bdf7f2e... M  resolv.conf

commit 163a31351672dea54bfe057802dcb2d1a2339794
Author: Charlie <root@localhost.(none)>
Date:   Sat Apr 25 01:07:08 2009 +0600

    Разрешил вход по паролю в sshd_config

:100644 100644 3547db4... e203db1... M  ssh/sshd_config

commit b22f34e46d80d9145cf636dee0f92a584433650b
Author: Charlie <root@localhost.(none)>
Date:   Sat Apr 25 01:04:16 2009 +0600

    Добавил DNS-сервер провайдера в resolv.conf

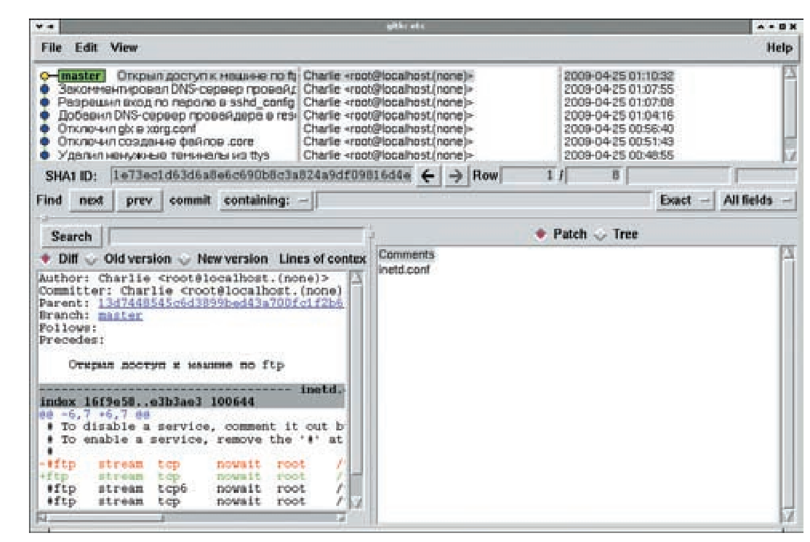
:100644 100644 bdf7f2e... fa7ce14... M  resolv.conf

commit 1331251683e397ceb923bc51681b69374401967a
Author: Charlie <root@localhost.(none)>
Date:   Sat Apr 25 00:56:40 2009 +0600
lines 1-35
```

› Всегда оставляйте осмысленные комментарии — потом будет легче вспомнить, что именно вы делали.

Другие решения

Идея хранения истории изменений конфигурационных файлов далеко не нова. Множество разработчиков в разные времена предлагали свои варианты ее реализации. Еще до появления децентрализованных систем управления версиями файлы **/etc** было принято хранить в CVS, что создавало некоторое неудобство ввиду ее требований к выделенному репозиторию. Не меньшей популярностью пользовались решения, основанные на файловых системах с функцией снимков состояния, таких как *backupfs* (http://sourceforge.net/projects/backupfs). Сегодня некоторые разработчики и даже компании предлагают комплексные решения по управлению конфигурацией серверов, среди которых можно выделить *Puppet* (http://reductivelabs.com/products/puppet/) – мощную, но сложную систему для управления конфигурациями, и *IsiSetup* (http://www.isisetup.ch/) – обертку для системы контроля версий *Subversion* (о которой мы говорили в **LXF118** и **120**), упрощающую многие задачи поддержания репозитория в актуальном состоянии. Несмотря на все это, большинство продвинутых пользователей предпочитают использовать для ведения истории конфигурационных файлов именно *Git*. LXF



› Помимо консольной, у *Git* есть и графическая ипостась.

Памятка пользователя Git

Иногда возникает необходимость не просто восстановить конфигурационный файл, но и узнать, какие изменения привели к нежелательным для системы последствиям. В этом вам помогут следующие несколько команд:

- **git whatchanged -5** – кратко о последних пяти изменениях;
- **git diff HEAD^** – различия между рабочим каталогом и репозиторием;
- **git diff "@{2009-05-05 9:00:00}"** – все изменения, произошедшие с определенного момента времени;
- **git diff "@{12 hours ago}"** – что изменилось за последние 12 часов.

В Интернет из консоли

Современные web-сайты радуют глаз визуальными эффектами, но основную смысловую нагрузку в Сети по-прежнему несет текст. **Игорь Штомпель** представит его без лишнего лоска.



Наш эксперт

Игорь Штомпель
Каждый день открывает в Linux что-то новое, и использует его где только возможно. Кстати, кто-нибудь установил gNewSense на FreeRunner?

На первый взгляд, идея неграфического рабочего стола может показаться безумием, каковым в свое время, наверное, выглядел оконный интерфейс *Xerox* на фоне привычных текстовых терминалов. А как же посещать любимые сайты, вести переписку по электронной почте, болтать с друзьями, рисовать, наконец? Ну, с последним, действительно, ничего не поделаешь (если только вы не владеете в совершенстве магией *ImageMagick* – кстати, мы писали о нем в **LXF116** и **117**), зато первые три задачи вполне решаемы. В этом нам помогут три программы: текстовый web-браузер *ELinks*, почтовый клиент *Alpine* и программа обмена мгновенными сообщениями *Finch*. Мы будем использовать Debian 5.0 Lenny, но все эти приложения наверняка отыщутся в репозиториях и вашего дистрибутива.

Прежде чем мы приступим к изложению, а наши читатели-ветераны – к написанию недовольных писем в редакцию, сделаем одно замечание: как обычно и бывает в Linux, перечисленные наименования не являются единственно верными. Кто-то может предпочесть связку *Lynx+Mutt+CenterICQ*, кому-то по душе оригинальный *Pine*. В общем, если вас что-то разочарует в указанных программах, но понравится сама идея, знайте: альтернативы есть.

ELinks: окно в Web

ELinks представляет собой текстовый web-браузер, поддерживающий протоколы HTTP/HTTPS, FTP, SMB и Finger, а также работу с локальными файлами. Прокси, cookie, таблицы, фреймы, вкладки, цвета и даже JavaScript – все это по плечу *ELinks*. Его также можно расширить сценариями на Perl, Lua и Guile.

Версия *ELinks*, входящая в основной репозиторий Debian 5.0 – 0.11.4-3, увы, имеет ряд проблем с русификацией интерфейса, хотя страницы отображаются нормально. Поэтому мы воспользуемся «нестабильным» *ELinks 0.12* – его можно загрузить с официального сайта (elinks.or.cz) или взять из репозиториев Squeeze. Скачайте пакеты **elinks_0.12** и **elinks-data_0.12**, а затем скомандуйте:



› ***ELinks*** **предоставляет множество параметров для настройки своего поведения.**

```
dpkg -i elinks-data_0.12~pre2.dfsg0-1_all.deb &&
elinks_0.12~pre2.dfsg0-1_i386.deb
```

и введите:

```
elinks
```

Программа предложит указать URL необходимого вам ресурса. Но для начала надо произвести предварительные настройки, поэтому нажимаем Esc для закрытия окна Go to URL [Открыть URL]. Кстати, в дальнейшем для его вызова можно использовать клавишу G.

Чтобы настроить *ELinks*, нажмите клавишу O: это приведет к появлению окна Option Manager [Менеджер опций]. Навигация осуществляется курсорными клавишами, для раскрытия списка используется пробел. Переместитесь на список Document [Документ] и откройте его, затем перейдите в Charset [Кодировка]. Выберите Default codepage [Кодировка по умолчанию], нажмите клавишу E для редактирования и в поле Value [Значение] укажите koi8-r для корректного отображения русских букв. Перейдите клавишей Tab на кнопку OK и нажмите Enter. Далее перейдите к списку User Interface [Интерфейс пользователя] и раскройте его. Выберите Language [Язык] и в поле Value [Значение] укажите Russian [Русский]. Затем нажмите кнопку OK, а в окне Option Manager [Менеджер опций] кнопку Save [Сохранить]. Теперь изменения сохранены, и отдельные элементы интерфейса *ELinks* будут переведены на русский язык.

Рассмотрим приемы работы с *ELinks*. Введите в адресной строке www.linuxformat.ru. Откроется главная страница сайта. Навигация осуществляется так: прокрутка страницы – PgUp и PgDn; перемещение по ссылкам – стрелки вверх и вниз; переход по ссылке – Enter. Нажатие клавиши t приведет к запросу URL и открытию сайта на новой вкладке (их «корешки» располагаются внизу экрана). Закрыть активную вкладку можно клавишей C.

При работе в Сети часто возникает необходимость добавить ту или иную страницу в закладки. В *ELinks* это можно сделать с помощью клавиши A. Для поиска по странице нажмите / (*ELinks* ищет подстроку, так что при вводе «сервер» будут выделены и «серверы»). Ну, а если вам интересен исходный код страницы, просмотреть его можно клавишей \.

Alpine: электронная почта

Alpine – это консольный почтовый и новостной клиент, основанный на *Pine*. Он поддерживает протоколы POP3/IMAP, SMTP и NNTP. В Lenny доступна версия 1.10, и чтобы установить ее, достаточно набрать:

```
apt-get install alpine
```

Почтовый клиент готов к работе. Войдите в систему как обычный пользователь и дайте команду:



› *Alpine* приветствует нового пользователя: горячие клавиши приведены внизу окна.

```
alpine
```

При первом старте программы будет показано приветствие. Чтобы попасть в главное меню, нажмите E. Для навигации можно пользоваться контекстными клавишами – они отображаются внизу экрана. Для главного меню это: вывести справку [HELP] –?, создать сообщение [COMPOSE MESSAGE] – C, просмотреть письма в текущей папке [MESSAGE INDEX] – I, выбрать папку для просмотра [FOLDER LIST] – L, адресная книга [ADDRESS BOOK] – A, настроить программу [SETUP] – S, выйти [QUIT] – Q. Традиционный способ – курсорные клавиши и Enter – можно применять в качестве альтернативы там, где это доступно.

Для начала настройки учетной записи нажмите клавиши S и C. Далее в Personal Name [Личное имя] наберите ваше имя (например, nameuser), а в User Domain [Домен пользователя] – имя почтового сервера (например, mail.ru). В поле SMTP Server (for sending) [Сервер SMTP (для отправки)] укажите имя сервера для отправки электронной почты (скажем, smtp.mail.ru). В Inbox Path [Путь к входящим] необходимо прописать путь к «папке» пользователя на почтовом сервере (например, pop.yandex.ru/user=ваше_имя@yandex.ru/pop3/). Поле Incoming Archive Folders [Папки архива входящих] дает возможность задать путь к каталогу, где будет храниться почта (например, **~/my_email**). Этих настроек достаточно, чтобы начать работать с электронной почтой.

Из главного меню переходим в FOLDER LIST [Список папок]. Программа запросит у вас пароль для указанной в настройках учетной записи, после ввода которого клиент проверит наличие на сервере новой почты. Затем ее можно будет просмотреть – папка INBOX [Входящие] в FOLDER LIST [Список папок]. Чтобы открыть сообщение, перейдите на него и нажмите Enter. Для возврата к списку сообщений используйте <. Сохранить сообщение для локального хранения (каталог, указанный в настройках в Incoming Archive Folders [Папки архива входящих]) можно, нажав S, а если требуется сохранить несколько сообщений за один раз, то нажмите **;** для отбора писем, а затем A (отобрать все письма). Выделенные сообщения будут помечены X. Далее – A, и чтобы сохранить письма, которые были выделены – S.

Для создания нового сообщения нажмите C в главном меню. Заполните все необходимые поля (To [Кому], Cc – копия, Attchmnt [Вложение; гласные, видимо, опустили для краткости], Subject [Тема]) и тело сообщения. Если требуется приложить к письму какой-либо файл, то для этого надо, находясь в заголовке сообщения, нажать комбинацию клавиш Ctrl+J и указать путь к нему внизу экрана в поле File to attach [Файл для вложения]. Нажав Ctrl+X, вы отправите письмо (потребуется ввести имя пользователя и пароль).

Finch: обмен сообщениями

Finch – это консольная версия *Pidgin*. Раннее программа была известна как *gaim-text*, и получила свое новое имя при переименовании *Gaim*. *Finch* – мультипротокольный пейджер, поддерживающий (в числе прочего) ICQ, XMPP/Jabber и IRC.

Установить и запустить *Finch* можно стандартной парой команд:

```
apt-get install finch
finch
```

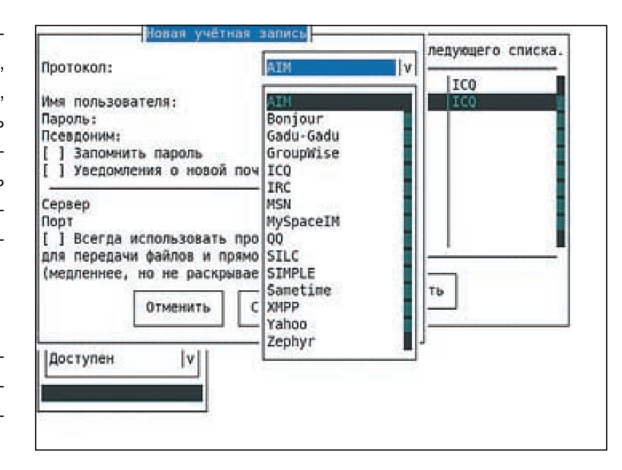
После этого клиент предложит добавить учетную запись; рассмотрим данный процесс на примере ICQ. Выберите протокол с помощью курсорных клавиш и перейдите (с помощью Tab) к полю Имя пользователя: в нем следует указать ваш номер ICQ. В поле Пароль надо ввести пароль. Поле Псевдоним, естественно, позволяет добавить псевдоним для учетной записи. Если вы хотите, чтобы *Finch* запомнил ваш пароль, перейдите к Запомнить пароль и клавишей пробел установите символ X в квадратных скобках; проделайте то же самое в следующем поле, если хотите получать уведомления о новой почте. Поля Сервер и Порт оставьте без изменений, а в поле Кодировка укажите CP1251. Кроме того, если вам необходимо всегда использовать прокси-сервер ICQ для передачи файлов и прямого соединения, установите символ X в соответствующем поле. Для сохранения изменений перейдите на кнопку Сохранить и нажмите Enter.

Программа выведет окно Учетные записи, которое позволяет включать/выключать их. Включите только что добавленную и нажмите Alt+C, чтобы закрыть окно. Вы увидите Список собеседников. Чтобы добавить в него запись, нажмите F11 (вызов контекстного меню) и выберите Добавить собеседника. В появившемся окне можно ввести номер ICQ (имя пользователя) и псевдоним, а также поместить собеседника в группу. Если вы используете несколько учетных записей, укажите, для какой добавляется контакт, и нажмите кнопку Добавить. Если новый собеседник не появился в списке, нажмите F10 и перейдите в Параметры, а затем выберите Показывать > Пустые группы и Показывать > Собеседники не в сети.

Перемещаться между контактами в списке можно с использованием курсорных клавиш, а открытие окна диалога с тем или иным собеседником осуществляется клавишей Enter. Навигация между окнами чата осуществляется комбинацией клавиш Alt+W или Alt+N (перейти на следующее окно) и Alt+P (перейти на предыдущее окно), закрытие окон – Alt+C.

Нажатие Alt+A выведет окно Actions [Действия], которое дает возможность настроить учетные записи, перейти в список собеседников, загрузить/выгрузить дополнительные модули, осуществить передачу файлов, настроить звуки и так далее.

На этом мы и закончим наше повествование. Конечно, мы затронули лишь базовые возможности указанных программ, но и из них понятно: консольные интернет-приложения обладают гибкими настройками и богатой функциональностью. Они вполне могут стать заменой своих графических аналогов для тех, кто предпочитает работать в терминале, или когда нет возможности запустить X-сервер. LXF



› *Finch* предоставляет вам на выбор объемный список протоколов… Кто-нибудь пользуется QQ?

Sox: Вскроем аудиофайлы

Шашанк Шарма препарирует имеющиеся у вас композиции с помощью *Sox*, универсального процессора командной строки.



Наш эксперт

Шашанк Шарма
Соавтор Beginning Fedora, и уже лет пять пишет о свободном ПО. Он также участвовал в Linux.com.

Как ни глупо это звучит, впервые я применил ***Sound Exchange*** **(*Sox*) для вырезания кусочка из саундтрека к фильму, чтобы сделать себе из него рингтон. А ведь** ***Sox*** **способен на большее, чем простое нарезание аудиофайлов. Простейшие операции, выполняемые с помощью программы — проигрывание музыки или преобразование форматов, но сложных функций для управления файлами в** ***Sox*** **просто бездна.**

Sox устанавливается по умолчанию в большинстве дистрибутивов, но там может быть не самая свежая версия программы. Текущий релиз – 14.3.0, и если ваша версия отлична от этой, первым делом удалите ее и установите последнюю. Избавиться от *Sox* можно при помощи графического менеджера пакетов вашего дистрибутива, или *Yum* и *apt-get*, если вы фанат командной строки. Пользователи Fedora могут удалить *Sox* командой

```
su -c “yum remove sox”
```

```
linuxlala@localhost:~/Music
File Edit View Terminal Tabs Help
[linuxlala@localhost Music]$ soxi Titoli.wav

Input File     : 'Titoli.wav'
Channels       : 2
Sample Rate    : 44100
Precision      : 16-bit
Duration       : 00:02:50.99 = 7540736 samples = 12824.4 CDDA sectors
Sample Encoding: 16-bit Signed Integer PCM

[linuxlala@localhost Music]$ sox Titoli.wav fade_in.mp3 fade 5 7540736s
[linuxlala@localhost Music]$
```

› **Продолжительность затухания звука по умолчанию равна продолжительности нарастания.**

Затем перейдите на страницу проекта по адресу http://sox.sourceforge.net и скачайте последний tar-архив. Все верно, это не пре-компилированные бинарники. Теперь запустите команду:

```
tar zxvf sox-14.3.0.tar.gz
```

для получения каталога **sox-14.3.0**.

Если вы хотите, чтобы *Sox* поддерживал MP3, добавьте перед его установкой библиотеки *libmad* и *Lame*. По ходу, также установите *libmad-level и lame-level*. Они обычно представлены в репозиториях большинства дистрибутивов, поэтому у вас не должно возникнуть проблем с их поиском. После установки, перейдите в только что распакованный каталог **sox-14.3.0** и запустите **./configure**, а затем **make** и **make install**.

Теперь на вашем компьютере есть последняя версия *Sox*, с поддержкой MP3, и все довольны – ну, разве что надулись фанаты OGG. Вы можете их утихомирить, отконвертировав вашу коллекцию MP3 в формат OGG командой

```
sox soundtrack.mp3 appeased.ogg
```

Вы можете проигрывать любой из этих форматов с помощью **play appeased.ogg**.

Выполнение операции

Теперь время перейти к более сложным задачам – а что может быть полезнее создания рингтона? Воспользуемся опцией **trim** для удаления из заданного аудиофайла всего ненужного, но сначала найдем «подопытного кролика», а затем применим к нему **soxi**. Эта команда читает заголовок выбранного файла и выво-

дит на экран полезные сведения. При использовании без ключей **soxi** выводит всю информацию, найденную в заголовке, а различными ключами вывод можно ограничить. К примеру, команда **soxi -d Titoli.wav** выведет длительность **Titoli.wav** в формате ЧЧ:мм:СС, а команда **soxi -r Titoli.wav** – частоту дискретизации.

```
[linuxlala@localhost ~]$ soxi Music/Titoli.wav
Input File : 'Titoli.wav'
Channels : 2
Sample Rate : 44100
Precision : 16-bit
Duration : 00:02:50.99 = 7540736 samples =
12824.4 CDDA sectors
Sample Encoding : 16-bit Signed Integer PCM
```

Для функции обрезки необходимо знать только Длительность; она выражается в формате ЧЧ:ММ:СС или в количестве звуковых отсчетов. Если вам интересны только начальные 40 секунд файла, выполните следующую команду:

```
sox Titoli.wav 40_seconds.wav trim 0 40
```

Для опции **trim** необходимы два параметра: start [начало] и length [длина]. Соответственно, 0 и 40 означают начальную позицию и продолжительность. В этом случае, мы берем первые 40 секунд от начала. Если вам нужен кусок с середины файла, замените 0 на соответствующее значение и укажите необходимую длину. Например, файл **middle.wav** изготовлен при помощи следующей команды:

```
Sox Titoli.wav middle.wav trim 130 150
```

Она создает 20-секундный клип, обрезая начало и конец исходного файла.

Затухание звука

Различают два типа меломанов: одни любят затухание и нарастание звука, а другие – нет. Если вы из лагеря первых, добавьте эффект постепенного нарастания звука с помощью

```
sox song.mp3 fade_in.mp3
fade 5 170
```

Здесь **5** – это длина нарастания, а **170** – время остановки (продолжительность песни) в секундах. Вы также можете наложить эффекты затухания и нарастания звука одной командой, согласно такому синтаксису:

```
sox song.mp3 fade_in_out.mp3 fade fade-in-length [stoptime
[fade-out-length]]
```

Таким образом, для создания пятисекундного эффекта нарастания и 10-секундного эффекта затухания подойдет следующая команда:

```
sox song.mp3 fade_in_out.mp3 fade 5 170 10
```

Затухание звука нельзя добавить к файлу, не указав время остановки, но его можно и не задавать, если вам нужен только эффект нарастания звука.

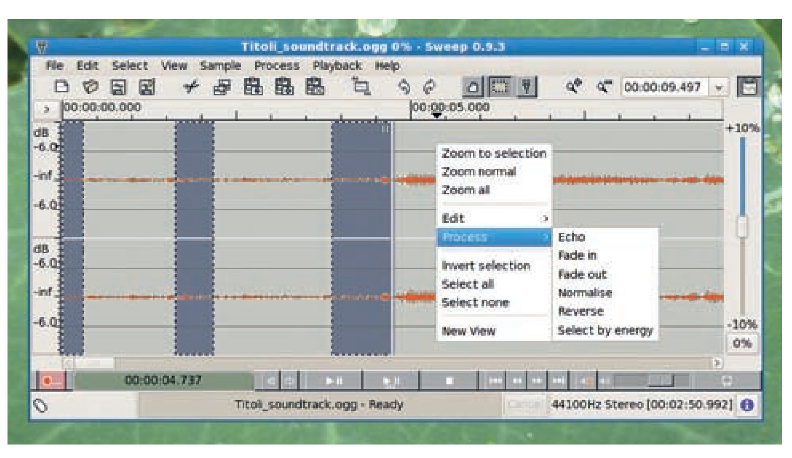
> «В Sox очень легко добавить эхо-эффекты к аудиофайлам.»

Отсчеты как мера длины

Если вам не подходит формат ЧЧ:ММ:СС, задайте время остановки числом отсчетов звука. В файле **Titoli.wav** например, содержится 7540736 отсчетов. Это число может быть использовано в команде так:

```
sox Titoli.wav fade_in.mp3 fade 5 7540736s
```

Буква **s**, приписанная к номеру отсчетов, очень важна: она сообщает *Sox*, что длина указана как число отсчетов, а не время окончания файла. Заметим, что можно использовать опцию **-s** с **soxi** для определения числа отсчетов в вашем файле. Разделив число отсчетов на частоту дискретизации файла, вы получите продолжительность трека в секундах.



› *Sweep* — еще одна графическая альтернатива *Sox*; его GUI позволяет выполнять сложные операции со звуком на разных частях аудиофайла одновременно.

В *Sox* легко добавлять к аудиофайлам эхо-эффекты. При работе с ними нужно задать команде *Sox* четыре значения: gain-in, gain-out, delay и decay. Delay – это разница во времени между исходным аудио и эхо-эффектом, который также называется отражением, а decay определяет громкость отражаемого звука по отношению к gain-in. Delay и decay указываются в миллисекундах (ms), а gain-in и gain-out описывают уровень громкости.

Ответь мне, эхо

Кроме стандартного эхо-эффекта, в *Sox* также существует функция echos. В ней исходный звук используется для создания начального отражения; это отражение и исходный звук затем используются для создания второго отражения, и т.д. Структура команды почти идентична основной команде echo: вы просто создаете больше пар delay/decay.

Теперь, пока вы не исказили свои аудиофайлы эхо-эффектами до неузнаваемости, вспомним команду **play**, о которой шла речь до этого. Вы можете применить следующую команду:

```
play Titoli.mp3 echos 0.8 0.7 500 0.5 600 0.8
```

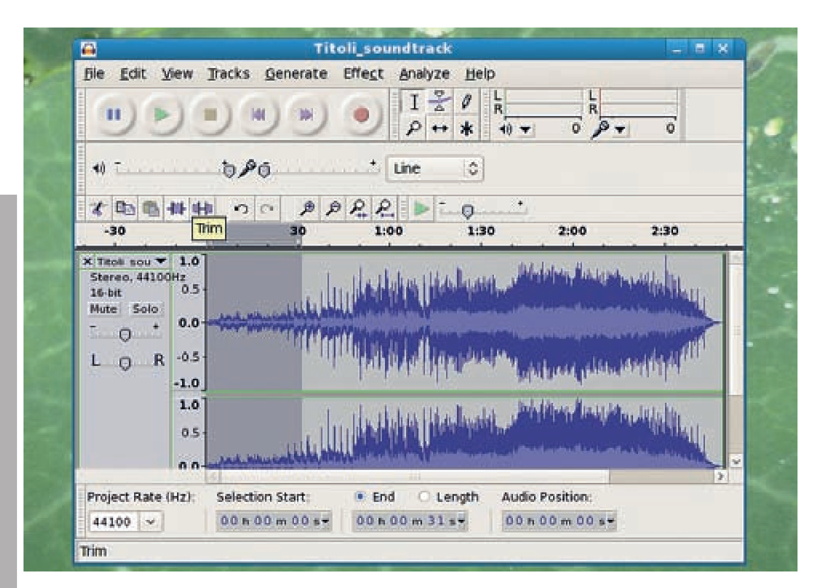
для прослушивания эффектов без изменения самого файла.

В этой статье мы остановились только на базовых командах, выполняемых с помощью *Sox*, но рекомендуем также обратить внимание на графического конкурента *Sox* – *Audacity*. LXF

Скорая помощь

В основном функциональность *Sox* осуществляется через команду **sox**, но команды **play** и **rec** можно выполнять сами по себе.

› *Audacity* — куда более сложный редактор аудиофайлов: он микширует треки и добавляет всевозможные эффекты через элегантный GUI.



```
#!/usr/bin/perl -w
use strict;
use Benchmark qw (:timethis);
my $count = 10;
timethis( $count, sub { '/path/to/myprogram' } );
#!/usr/bin/perl -w
use strict;
use Benchmark qw (:all);
cmpthese( 10, {
        a => sub { '/path/to/oldcode' },
        b => sub { '/path/to/newcode' },
    } );
use Benchmark qw (:hireswallclock);
#!/usr/bin/perl -w
use strict;
use Time::HiRes;
sub time_print;
print time_print . ":";
# code here
print time_print . ":";

print time_print . "\n";

sub printtime {
            my ($t1, $t2) =
Time::HiRes::gettimeofday;
    my $time = "$t1." . sprintf("%05d",
$t2/10);
    return "$time";
}
use strict;

my $datafile = "testout.txt";
my @timearray;
my $count = 0;
```

Оценка быстродействия и профилирование



Наш эксперт

Джульетта Кемп весьма ленива насчет оптимизации кода, и предпочитает действовать направленно. Вот почему она любит профилирование.

Бывает, что код до ужаса тормозит. **Джульетта Кемп** покажет, зачем использовать черную магию тестирования производительности.

Работаете ли вы с оборудованием или с программным продуктом, на каком-то этапе скорость обязательно станет проблемой. Задача это обширная, и мы ограничимся ситуацией, когда медленно работает код, и рассмотрим, как его ускорить. Меры могут быть разными, от переписывания программы до покупки нового компьютера, но прежде всего необходимо найти основную причину и узкое место. Тут-то и пригодится тестирование производительности — и весьма важно сделать его правильно, прежде чем переходить к реформам. При сравнении различных участков кода этот процесс также иногда называют профилированием.

Тестирование производительности важно рассматривать как часть процесса усовершенствования кода. Для начала создайте код и убедитесь, что он правильно работает – а уж потом оценивайте, достаточно ли он быстр. В конце концов, нет особого смысла ускорять код, если он и так справляется: вашу энергию можно употребить на другие дела.

```
Terminal — bash — 80x24
glade:testscripts juliet$ cat bmwrapper.pl
#!/usr/bin/perl -w
use strict;
use Benchmark qw (:all);

my $count = 100;
cmpthese( $count, {
                a => sub { `./testbm2.pl` },
                b => sub { `./testbm3.pl` },
        } );

glade:testscripts juliet$ ./bmwrapper.pl
     Rate    b    a
b  476/s   -- -71%
a 1667/s 250%   --
glade:testscripts juliet$
```

› Вывод скрипта-обертки, использующей *Benchmark.pm*.

Повторяющийся процесс

А вот если ваш код тормозит, пора заняться тестированием. Затем используйте результаты (вместе с анализом и профилированием) для определения наилучшей точки приложения усилий. Реализовав исправления, тестируйте снова, чтобы увидеть, достаточно ли принятых мер. Если нет, переходите к следующей области и продолжайте работу.

Важно убедиться, что ваши числа состоятельны (то есть вы понимаете, что получили) и сосредоточить усилия на том участке, где эффект будет максимален. Явно не стоит двое суток биться над алгоритмом, если задержка обусловлена записью на диск.

Начальная оценка

Первым делом протестируйте код в его текущем состоянии. Практически всегда желательно запустить программу несколько раз подряд, чтобы выборка был репрезентативной. Это позволит усреднить длительность прогона, которая может меняться в зависимости от нагрузки процессора. Лучше всего применить скрипт-обертку, вроде такого:

```
#!/usr/bin/perl -w
use strict;
use Benchmark qw (:timethis);
my $count = 10;
timethis($count, sub {`/путь/к/программе`});
```

Он написан на Perl, но поскольку для вызова тестируемого кода используются обратные апострофы (`), его можно использовать для хронометража программы на любом языке, запускаемой из командной строки. В тексте скрипта вы заметите **$count** – это количество итераций. Другой формат – знак минус и минимальное число секунд времени процессора на прогон: например, ввод **-5** приведет к запуску как минимум на 5 секунд процессорного времени.

Benchmark.pm, использованный выше модуль Perl, удобен также и для тестирования производительности программ не на Perl: просто применяйте апострофы для вызова кода из Perl-скрипта, как было сделано выше. *Benchmark.pm* обычно поставляется с достаточно свежей версией Perl, но может быть установлен и через *CPAN*. Подправив код, снова воспользуйтесь *Benchmark.pm*, чтобы выяснить, намного ли код стал быстрее:

```
#!/usr/bin/perl -w
use strict;
use Benchmark qw (:all);
cmpthese( 10, {
        a => sub {`/путь/к/старомукоду`},
        a => sub {`/путь/к/новомукоду`},
 })
```

Отметим, что **cmpthese** сравнивает количество прогонов кода в секунду, а не время выполнения. Однако это может быть полезнее, чем абсолютное время, на которое влияет планировщик ядра и другие системные функции. Скрипт-обертка также подавляет любой вывод исследуемого кода в **stdout**. По умолчанию, **cmpthese** работает до ближайшего целого числа секунд, но можно включить высокоточный таймер при помощи

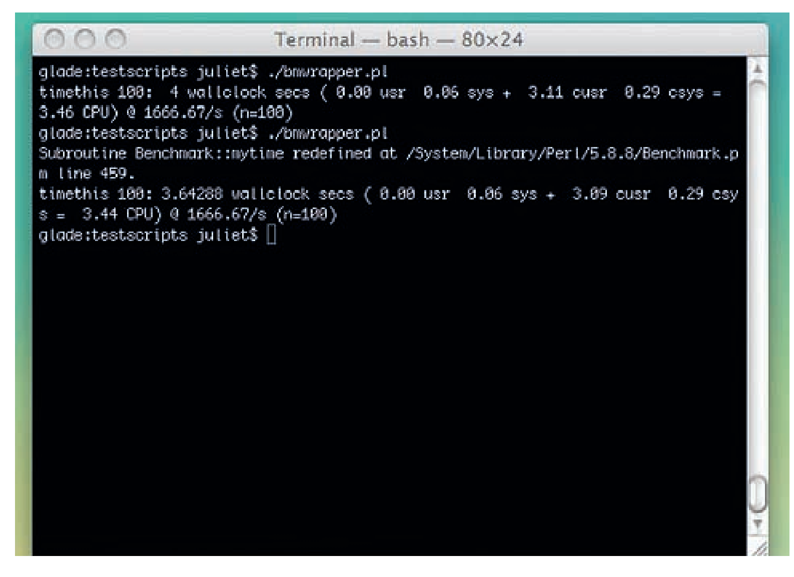
```
use Benchmark qw (:hireswallclock);
```

чтобы получать значения времен в микросекундах.

> «Потом Benchmark.pm сообщит, насколько ускорился код.»

Ведем записи

Обертки выводят все полученные значения времени на консоль, но вывод можно перенаправить в файл (используйте **wrapPer.pl >> benchmark.txt**, чтобы результаты каждого прогона добавлялись к нему, не перезаписывая старых), для отслеживания прогресса.



› Первый результат содержит стандартные времена *Benchmark*, а во втором используется hireswallclock. Отличие — в единицах измерения: hireswallclock выдает результаты в микросекундах, а не в секундах.

Учтите, что, приступая к редактированию, всегда следует держать под рукой копию старого кода. Это позволит выполнить сравнение (как показано выше) и проверить, каких результатов вы достигли, поскольку при оптимизации всегда можно внести ошибку. К тому же это будет той страховочной сеткой, что спасет вас, если вместо улучшения все рухнет. Вообще-то не мешает использовать систему контроля версий и регулярно фиксировать в ней изменения: эти усилия окупаются сторицей благодаря возможности отката на заведомо работоспособную версию, когда что-то идет не так.

Итак, настал момент запуска первого набора простых тестов. Помните, что во время тестирования не следует запускать другие задачи, иначе результаты собьют вас с толку. По возможности следует также выполнить скрипт-обертку, сам дающий усредненные результаты, несколько раз, и усреднить еще и их, добиваясь более показательного значения. Убедитесь, что ничего не загружаете: это может сказаться на скорости обработки или ввода/вывода. Желательно выполнить столько прогонов, на сколько у вас хватит времени или терпения: чем больше прогонов, тем точнее будет результат.

Скорая помощь

Даже если у вас в данный момент нет проблем с быстродействием, желательно выполнить пару тестов производительности для выявления средних значений. Затем, если вам покажется, что работа стала медленнее, чем раньше, вы сможете сравнить данные и понять, так ли это, или просто вы стали менее терпеливы.

Профилирование: расчленяем код

По получении оценки, следующий шаг – логическое деление кода. Просмотрите его и выделите разделы, на которые его можно разбить. Вот некоторые предложения:

- раздел работы с диском (например, чтение или запись);
- создание и/или заполнение структур данных;
- алгоритмы и вычисления;
- какая-либо работа с сетью (но, вероятно, здесь ускорение будет вне вашего контроля).

На данном этапе для компилируемых языков следует, вероятно, слегка переделать ваш код или разбить его на части. Вывод на диск убрать довольно просто – для этого иногда достаточно пару раз что-то закомментировать; но ввод с диска выкинуть сложнее, поскольку для получения результата алгоритму нужны данные. Вместо этого можно поступить наоборот, убрав все, кроме чтения; вычтите среднее время таких прогонов из среднего времени работы полной версии, и получите среднее время работы алгоритма.

»

Для примера допустим, что имеется три основных раздела кода: чтение данных, проведение вычислений и запись новых данных. Создайте три версии вашего кода:

A: Полная версия

B: Только чтение данных (без расчетов и вывода).

C: Чтение данных и вычисления (без вывода).

Пусть тестирование трех этих версий дало следующее: версия A выполняется 4 секунды, B – 1 секунду и C – 3,5 секунды. То есть чтение данных занимает 1 секунду (из B). Вывод данных занимает 0,5 секунды (разность между A и C). И, следовательно, вычисления идут 2,5 секунды. Вы можете найти это по формуле **A – B – вывод данных = время работы** (в нашем случае 4 – 1 – 0,5 = 2,5). Вот время, требуемое для работы без ввода и вывода данных.

> «Весьма важно найти узкое место вашего кода.»

Вы можете использовать данный метод при хронометраже других участков кода, которые трудно отделить.

Альтернатива – выводить в контрольных точках системное время. Скрипт Perl, приведенный ниже, сгенерирует при запуске по одной строке на прогон кода:

```
#!/usr/bin/perl -w
use strict;
use Time::HiRes;
sub time_print;
print time_print . “:”;
# здесь код
print time_print . “:”;
# еще код
print time_print . “\n”;

sub printtime {
  my ($t1,$t2) = Time::HiRes::gettimeofday;
  my $time = “$t1.” . sprintf(“%05d”,$t2/10);
  return “$time”;
```

и выдаст нечто подобное:

```
1240905933.05204:1240905934.05249:1240905935.05264:1240
905936.05312
```

Значения в этих парах разделяются точками, а пары отделены двоеточием. Значения в паре – в секундах, прошедших с начала эпохи, и в микросекундах; первое – время запуска скрипта, а второе – время завершения. Использование модуля *Time::HiRes* позволяет выполнять подсчет в микросекундах, а не в целых секундах.

```
Terminal — bash — 80×24
glade:testscripts juliet$ ./testbm.pl > testout.txt
glade:testscripts juliet$ ./testbm.pl >> testout.txt
glade:testscripts juliet$ ./testbm.pl >> testout.txt
glade:testscripts juliet$ less testout.txt
glade:testscripts juliet$ ./datahandle.pl
Section 1 average = 1.00025002161662
Section 2 average = 1.00083009401957
Section 3 average = 1.0011899471283
Section 4 average = 1.00453654925028
Section 5 average = 1.00018334388733
Section 6 average = 1.00024668375651
Section 7 average = 1.00027338663737
Section 8 average = 1.00036334991455
Section 9 average = 1.00018334388733
Section 10 average = 4.66505686442057e-05
glade:testscripts juliet$
```

› Запуск обработчика вывода для выполнения базового профилирования участков кода.

Вычисление времени

Альтернативой для хронометража при использовании Perl и *Benchmark* является вычисление разности времен при помощи

```
my $t1 = new Benchmark;
# code
my $t2 = new Benchmark;
my $td = timediff($t1, $t2);
print “Первый раздел потребовал $td \n”;
```

Однако такой результат труднее обрабатывать автоматически.

Обработка данных

Выполните код несколько раз, перенаправляя вывод в файл, а затем воспользуйтесь следующим скриптом Perl для его обработки:

```
#!/usr/bin/perl -w
use strict;

my $datafile = “testout.txt”;
my @timearray;
my $count = 0;

open DATA, $datafile;
while (<DATA>){
  my @time = split /:/;
  push @{ $timearray[$count] }, @time;
  $count++;
}
close DATA;

my @result;
for my $rowref ( @timearray ) {
  my @row = @$rowref;
  for my $i ( 0 .. ($#row – 1) ) {
     $result[$i] += $row[$i+1] – $row[$i];
  }
}

for my $i ( 0 .. $#result ) {
  print “Section “ . ($i+1) . “ average = “ . $result[$i] / $count . “\n”;
}
```

Это приведет к выводу результата (*как показано на экранном снимке внизу слева*), но учтите, что скрипт-тест выполнялся три раза, а этого мало для правильного профилирования.

Теперь вы должны иметь представление о длительности выполнения вашего кода в целом, а также его отдельных частей. Следующий шаг – ускорение.

Решение проблем

Как указывалось выше, важно найти в вашем коде узкое место, чтобы сосредоточить усилия там, где от этого будет больше всего пользы. Не исключено, что вы распознаете проблемный кусок кода, просто поглядев на цифры; а если не получится, постройте графики с помощью *OOo*, *KChart*, *Guppi* или *Graphviz*. Кто предпочитает работать в консоли, может взять **gnuplot**.

Если ваш код полон операций ввода/вывода, весьма вероятно, что они-то и есть узкое место. Скорость можно увеличить при помощи следующих шагов:

» Обеспечьте работу локально, а не в сети. Завершив работу с файлами, перекинуть их на удаленные диски вы всегда успеете.

» Приобретите новую периферию. Возможно, ваша старая просто уже не на высоте. Вы можете бросить ее на те участки, где скорость менее важна.

» Выполняйте операции ввода/вывода пакетно. Например, считайте данные все разом, а затем уже обрабатывайте их в памяти. Если код выполняется несколько раз, то создайте процедуру чтения данных один раз в самом начале.

» Понизьте «уступчивость» (nice-значение) своего процесса.

Для просмотра текущих параметров диска можно применить **hdparm** (например, **hdparm -v /dev/hda**). Можно увеличить скорость работы накопителя лобовой атакой – включить прямой доступ к памяти (Direct Memory Access, DMA) (**hdparm -d1 /dev/hda**); затем вновь оцените быстродействие и посмотрите, сработало ли это. Также можно изменить значение поддержки ввода/вывода (I/O support) опцией **-c3**, это также способно слегка улучшить быстродействие. Поэкспериментируйте с другими значениями **hdparm**, но будьте осторожны: некоторые из них могут быть опасны. Перед выполнением действий почитайте man-страницу. При желании сохранить произведенные изменения, выполните **hdparm -k /dev/hda**.

Подгонка файловой системы

Вы можете ускорить доступ к файлам, отключив ведение временных отметок при каждом обращении к ним – то есть не делая запись в каталог при каждом открытии файла. Для этого отредактируйте **/etc/fstab**, добавив **noatime** к списку опций в четвертом столбце для каждой ФС, которую вы хотите изменить, а потом все перемонтируйте командой **mount -a**. Однако отсутствие записей о доступе к файлу способно вызвать проблемы в некоторых программах: почтовых или резервного копирования. Хотя можно создать для своего приложения отдельный раздел и использовать **noatime** только в нем.

Strace и ltrace

Strace выводит список системных вызовов, выполненных программой. Обычно для тестирования это перебор, но тут есть пара полезных опций. Например, **-c** записывает время каждого системного вызова, **-r** выводит относительное время каждого системного вызова, а **-t** – абсолютное (для включения микросекунд используйте **-tt**). **-T** выводит время, потраченное на системный вызов. С опцией **-e** будут отслеживаться только определенные наборы системных вызовов.

ltrace выполняет то же самое, но для библиотечных вызовов. И вновь **-c** посчитает время вызовов и возвратит сводку, а **-t** и **-tt** покажут время суток при запуске каждой строки.

› ***Strace***, **выполняемый для одного из скриптов-примеров на Perl.**

Параллельная обработка

Если алгоритм больше уже не улучшается, попробуйте распараллелить обработку. Возможно, для этого придется изменить подход к написанию кода, разветвив свои действия на несколько параллельных потоков. Например, при выполнении анализа объемного набора данных, части которого не взаимодействуют друг с другом, попытайтесь запустить один и тот же алгоритм сразу на нескольких машинах, с разными частями данных. В данной статье нет места для детального рассмотрения, но начать можно с учебников, имеющихся в сети. Увы, не все задачи распараллеливаются!

Если вы решили купить новое оборудование, то прежде чем тратить деньги, попробуйте одолжить на время чью-нибудь машину с искомыми характеристиками и провести несколько тестов производительности приложения. Лучше знать заранее, чего вы добьетесь.

Если узкое место – ваш алгоритм, то имеется много ресурсов для их анализа. Например, хорошо изучены алгоритмы сортировки, что позволяет выбрать из них оптимальный для ваших данных – например, сортировка при помощи двоичного дерева в общем случае лучше, чем пузырьковая. Впрочем, это выходит за рамки данной статьи – в сети есть множество информации для исследования.

Имейте в виду, что оптимизация алгоритма – работа крайне тяжелая, и следует убедиться, что итогом потраченного времени не будет жалкая мелочевка. Но иногда правильный выбор алгоритма приводит к радикальным переменам.

Скорая помощь

Опция **relatime** – это улучшенная версия **noatime**; если вы используете Ubuntu, не мешает ее попробовать.

Заключение

Для получения более подробной информации о том, где тормозит ваш Perl-код, можно запустить *Devel::NYTProf* или его предшественника *Devel::DProf* – это всеобъемлющие профилировщики с хорошей документацией. Используя данные инструменты, вы сможете найти проблемные места конкретных модулей и обдумать пути их исправления. Не забудьте потом опять выполнить профилирование, чтобы проверить, есть ли сдвиг.

Если вы используете Perl или другой скриптовый язык, и способов ускорения кода не обнаруживается, а быстродействие совершено неприемлемо, то, возможно, надо перейти на компилируемый язык – типа C, C++ или Java. Однако, прежде чем погрузиться в это и полностью переделать код, следует хорошенько убедиться, что это действительно стоит вашего времени и затраты усилий. Еще раз рассмотрите результаты тестирования и убедитесь, что сделали все возможное для ускорения и дальше здесь ехать некуда.

По завершении оптимизации выбранного раздела кода весьма важно проверить, что ваш новый код работает так же, как старый. Лучший способ сделать это – модульные тесты, и это еще один повод сохранить старый код, чтобы сравнить две версии. Во время оптимизации вполне вероятно насажать ошибок. Делайте заметки обо всех изменениях, чтобы не повторять ошибок в будущем и не забыть, что вы делали, на случай выполнения схожих операций.

На данном этапе надо снова протестировать производительность! Выясните, сколько времени вы сэкономили (не позабыв отметить, сколько рабочего времени ушло на поправки) и нельзя ли выиграть еще. Если нужно избавиться еще от пары секунд, посмотрите, может ли помочь еще что-то? Или необходимо переделывать весь код заново?

Процесс тестирования производительности и оптимизации кода – весьма увлекательное занятие, если приступать к нему с ясным представлением о цели. Однако это может стать источником досады, если вы поленитесь сперва все тщательно обдумать. Убедитесь, что ваш случай – первый, и наслаждайтесь переработкой своих алгоритмов и перерасходом на сверкающее новое оборудование. LXF

Что за штука... APML?

Марко Фиоретти расскажет, как «объяснить» web-сайту, что вас интересует.

Ура – еще одну аббревиатуру... забуду! Что она означает-то?
Attention Profiling Markup Language («Язык систематизации объектов внимания»), он позволяет web-службам отслеживать интересы и выявлять потребности пользователей.

Это насчет онлайн-рекламы? Она меня не волнует – я ее фильтрую.
Подавление web-рекламы – верный способ пройти мимо web-сайта с уникально интересным для вас контентом. Вам это надо? К тому же реклама – лишь одна из областей, которые способен преобразить APML.

Хм... Ну и что еще преображает APML?
Представьте себе Интернет в виде ярмарки, где каждый продавец забегает перед вами и кричит: «Взгляните на меня, купите мой товар, купите, купите, купите!»

> «Зная ваши интересы, сайт не всучит вам то, что вы игнорируете.»

Представить нетрудно – потому-то я и поставил фильтр...
Понимаю. Тогда вообразите, что у вас при себе плакат с надписью: «Даже не смотрите в мою сторону, если только вы не продаете китовый ус – сегодня мне только он и нужен». А потом напрягитесь и вообразите, что все торговцы повинуются надписи!

Вот уж чудо из чудес! Неужели APML это удастся?
Да, APML обещает стандартный, в значительной мере автоматизированный способ «дать понять» при входе на сайт, что именно вы предпочитаете.

Откуда же APML про это узнает?
Будут отбираться подсказки, предоставляемые вашими путешествиями по Интернету: закладки; часто посещаемые web-сайты, категории в вашем блоге или учетной записи на Flickr; темы на форумах, которые вы просматриваете чаще всего; объекты ваших поисков и т. д.

И что APML сделает с этими данными?
Зная ваши интересы, любой посещаемый вами сайт уже не подсунет вам то, что вы наверняка проигнорируете. APML предлагает метод сбора и систематизации подобных сведений – в полуавтоматическом режиме под вашим контролем – в форме, ясной и понятной всем сайтам.

Намекаете, что APML создан не только для рекламы?
APML – это универсальный способ «объяснить» удаленному компьютеру, что именно вас интересует. А затем этот компьютер распорядится полученной информацией в соответствии с заложенной программой. Нетрудно догадаться, что польза такого подхода не ограничивается персонализированной рекламой. Например, можно повысить эффективность просмотра новостей, онлайн-шопинга и интернет-поиска.

Давайте начнем с шопинга.
Давайте. Скажем, Пол обожает скутеры фирмы Vespa: он набил свой браузер закладками на страницы с любыми упоминаниями о Vespa, активно участвует в форумах фанатов Vespa, и так далее. И вот он заходит на сайт дилера подержанных скутеров. Если на нем заранее узнают о пристрастии Пола к Vespa, то соответствующие сведения всплывут на первые же строки домашней страницы.

Круто. Вы сказали, с новостями так тоже можно?
Конечно. Вы сами выбираете ленты RSS для загрузки, но вынуждены скачивать их целиком, даже если отслеживаете постоянно всего одну-две темы. Используя APML, любой RSS-клиент или новостной портал выдаст волнующие вас сообщения в первых строках списка.

А что с интернет-поиском?
Если вы войдете в поисковую машину или базу данных библиотеки, применив APML, то его данные, как и слова, введенные в строке поиска, будут использованы для получения более подходящих результатов.

Теперь о технических деталях. Как работает APML?
Если коротко, то ваши Attention Data («Данные об интересах») сохраняются в вашем личном Attention Profile («Профиле интересов»).

Яснее не стало. Что такое Attention Data?
Это набор переменных, описывающих вашу онлайн-активность на основе сведений о часто посещаемых вами сайтах, RSS-лентах, блогах или постах, закладках, фотографиях или видеозаписях, которые вы просматриваете или публикуете на Flickr или YouTube, музыке, которую слушаете на Last.fm, и т. д., и т. п.

А что такое Attention Profile?
Это систематизированная коллекция данных об интересах, каждому разделу которой присваивается рейтинг – числовое значение – и временная метка. Порядок, в котором ваши интересы регистрируются в данном профиле, позволяет сайту «понять», что вам в данный момент нужно больше всего.

Каков формат файла APML?
XML. Когда вы обращаетесь к службе или приложению, поддерживающему APML, происходит создание или обновление XML-файла формата APML. Затем вы передаете его web-сайту для повышения качества обслуживания.

Вроде начинаю понимать. Что же конкретно находится внутри APML-файла?
Сводка ваших скрытых и явно выраженных интересов, их источники и авторы, а также численные значения рейтинга

этих интересов (в виде процентов). И еще временные отметки для каждого поля.

Не слишком ли это статично? Я не всегда покупаю одно и то же.

Так для этого и нужны временные отметки и рейтинги! Если вы задумали сменить автомобиль, то перед покупкой нового вы, вероятно, накинетесь на соответствующие сайты. В этот период значение, присвоенное вашим интересам по части приобретения автомобилей, будет постоянно расти, а временные отметки в этом разделе будут обновляться по несколько раз в день. Высокие рейтинги и свежие временные отметки – явное свидетельство, для любой web-службы, резкого повышения интереса к данной сфере, и повод для направления в ваш адрес соответствующего контента.

Допустим, я уже купил автомобиль…

Скорее всего, вы прекратите посещение сайтов автодилеров. Временные отметки начнут устаревать, а рейтинг соответствующего раздела будет снижаться – в результате APML-службы перестанут направлять вам «автомобильную» информацию.

Насколько распространена эта штука? Многие ли онлайн-службы уже используют APML?

В число web-сайтов, использующих APML или заявивших о его использовании, входят Google, Digg, Bloglines, NewsGator, LastFM и Delicious. Существуют модули WordPress, позволяющие сформировать файл APML из любого блога (http://wordpress.org/extend/plugins/apml). Есть порталы, например www.particls.com, http://engagd.com и www.cluztr.com, специально созданные для использования и обмена данными Attention Data.

А я и не знал об этом ничего… Как думаете, станет ли APML всеобщим стандартом, или ему суждено раствориться в безвестности?

Интересный вопрос – сейчас на него ответить непросто. Много шума вокруг APML было в конце 2007 – начале 2008 года. Потом настало затишье. Это может означать и то, что механизм вполне готов к использованию, и то, что часть планов забуксовала.

Но что могло пойти не так?

Чтобы APML реально заработал, он должен быть целостным, актуальным и почти невидимым для конечного пользователя. Эти цели противоречивы, их трудно достичь одновременно. Еще один аспект – конфиденциальность информации.

Да, о ней мы как-то забыли. Может ли APML нарушить ее?

Ну, если вы завсегдатай социальных сетей или постоянный пользователь служб web-почты (Google, Hotmail, Yahoo…), то не все ли равно, в каком формате хранятся всем известные данные? Проблемы с конфиденциальностью зависят от использования сведений их получателями – неважно, с APML или без.

А если поподробнее?

Заявляя о себе – будь то через файл APML или через резюме на клингонском языке – вы в любом случае не контролируете действия получателя, включая мошенничество с вашими данными. Кроме того, сайтам, которые запрашивают APML-файлы, ничто не мешает добывать сведения о вашей личности и другими способами. Тут уж семь бед – один ответ.

Есть ли другие проблемы?

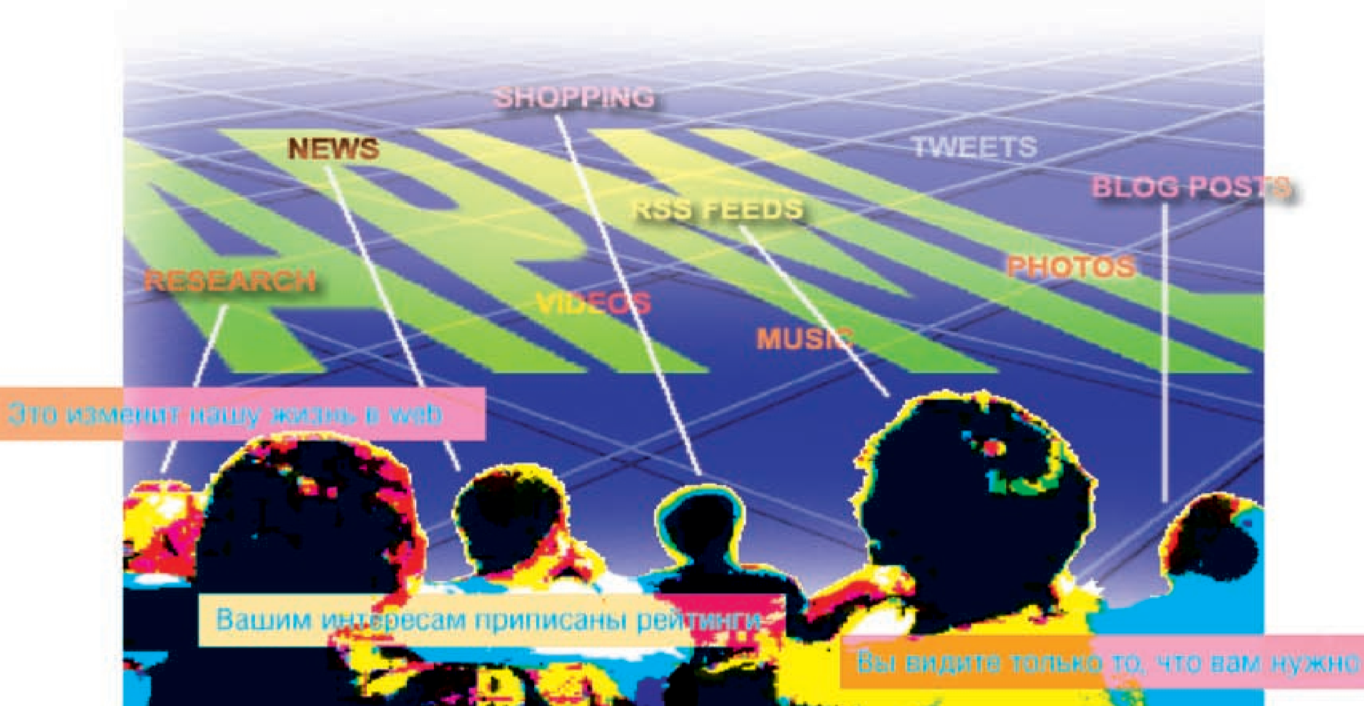
Да. APML – это один из способов видеть только то, что нам необходимо. Игнорировать то, что нас в данный момент не интересует, чертовски удобно, но не всегда умно. Например, из-за фильтрации новостей вы можете пропустить важные изменения в мире, и выпадете из реальности. Короче, должен быть компромисс, и каждый выбирает его сам.

Кто занимается разработкой APML? Могу ли я помочь?

Адрес рабочей группы APML– www.apml.org. Кроме Википедии, группа представлена на Facebook и в Google Groups (http://groups.google.com/group/apml-public/topics).

Где можно прочесть побольше?

Ответы на часто задаваемые вопросы можно найти на http://groups.google.com/group/apml-public/web/apml-faq, а спецификация APML находится по адресу http://apml.pbworks.com. Если же вы просто хотите подробнее узнать о целях и задачах APML, то стоит прочесть следующие статьи: http://tinyurl.com/3d2h9d, www.cleverclogs.org/2007/10/basics-of-atten.html и http://eliasbizannes.com/blog/2007/10/explaining-apml-what-it-is-why-you-want-it. LXF





По рецептам доктора Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.

Байтом меньше в глобальном потеплении

Будучи читателем этого журнала, вы, конечно, знаете, что нужно заботиться о нашей чудесной голубой планете, повторно перерабатывая отходы. Один из аспектов этой заботы, который давно меня беспокоит – это биты информации, которые мы выбрасываем за ненадобностью.

И я рад сообщить о новой инициативе по уменьшению объема цифровых отходов. Согласно принятым мерам, к 2015 г. мы сможем достичь его снижения на 75 %, несмотря на текущие темпы развития цифрового рынка. Например, с 1 апреля 2012 года уже нельзя будет создать новый TCP, UDP или IP-заголовок. Нужно будет повторно использовать старые протоколы, соответствующим образом изменив поля.

Для этого местные власти установят в вашем квартале серверы переработки. Потребители обязаны будут делить свой цифровой мусор на отдельные файлы – кадры Ethernet, всевозможные заголовки протоколов, публичные ключи SSH, истекшие сертификаты X590 и т.д – и они будут отправляться на переработку. Все, что не попадает в определенную категорию, будет рассортировываться на единицы и нули и также перерабатываться.

Если этого не сделать, биты будут постепенно исчерпаны, и дойдя до последней единицы, цифровой мир резко встанет. А может, последним будет ноль – сказать трудно.

Маркетологи, несомненно, попробуют вскочить на подножку уходящего экопоезда, и мы увидим в магазинах коробки с приложениями, снабженными зеленой наклейкой в углу и надписью «Эта программа на 90 % состоит из повторно используемых битов».

Славные перемены, что и говорить.

В данной колонке 12536 бит, и это 15 % от рекомендованного норматива дневного потребления.

Простая интеграция ПК

Likewise Open Забудьте о проблемах подключения сервера Linux к домену Active Directory.

Мир интеграции многим обязан команде разработчиков Samba. Однако настроить *Winbind*, *Samba* и PAM на компьютере с Linux, чтобы подключить его к домену Active Directory, все еще непросто. Если вы думаете заняться этим, обратите внимание на продукты от Likewise. Простой вариант для начинающих называется *Likewise Open* и позволяет подключить компьютер с Linux к домену за один шаг из командной строки или через графический интерфейс. Программа выполняет все необходимые настройки для того, чтобы поддерживающие PAM службы хоста могли аутентифицировать пользователей домена Active Directory, учитывая их членство в группах AD.

Прежние годы

Много лет назад я пытался читать курс лекций по интеграции Unix и Windows NT. Он явно опережал свое время – тогда не существовало хороших решений для разделения ресурсов или SSO в смешанной среде Unix, Linux и Windows.

Ослабьте привязь

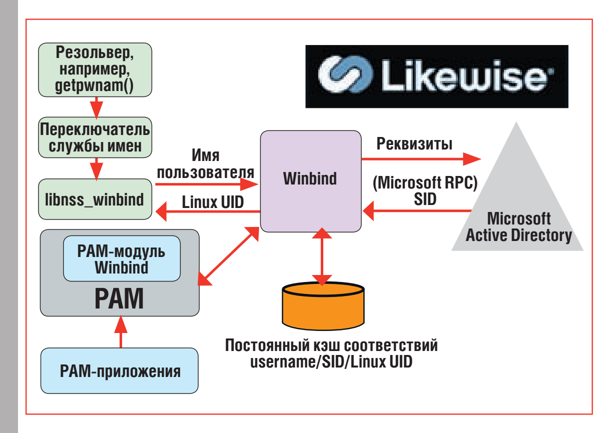
Добавив компьютер в домен, вы сможете войти в систему как любой пользователь Active Directory. Имя пользователя нужно будет ввести в формате **'domain\username'**, например

```
ssh 'example\chris'@hostname
```

Likewise Open не требует расширений для схемы Active Directory. Linux-часть подключается к библиотекам *resolver* и PAM, а Windows-часть действует как клиент Kerberos 5 и LDAP для аутентификации и авторизации соответственно.

Потенциальная проблема *Winbind* состоит в автоматическом назначении идентификаторов пользователя и группы Linux (UID и GID) пользователям Active Directory. При желании управлять этим, воспользуйтесь *Likewise UID-GID Module* (он не бесплатен). Данный модуль расширяет Active Directory, так что пользователям и группам Linux можно назначать идентификаторы по своему усмотрению.

Более подробную информацию ищите на www.likewise.com.



› *Winbind* «перекидывает мостик» между Linux и Active Directory.

Реально безопасная оболочка

SSH Напуганы записями о неудачных попытках входа в систему по SSH, кишащими в файлах журналов? Задрайте люки по плану о шести пунктах.

Если ваш SSH-сервер или компьютер открыт для доступа через Интернет, то число неудачных попыток входа в систему будет достигать сотен и тысяч в день. Феноменально? Кто не верит, загляните в файлы журналов. И что тут можно предпринять, кроме как струхнуть? Рассмотрим шесть способов повысить защищенность входа в систему через SSH.

1 Надежные пароли

Надежные пароли – первый рубеж обороны. Изучите критерии надежности пароля (и обучите своих пользователей, если они у вас есть). В Интернете можно найти массу материалов, но по сути, надежные пароли должны быть достаточно длинными, содержать смесь букв, цифр и спецсимволов, и не походить на слова из словаря. Если у вас есть пользователи, которым разрешено брать себе пароли самостоятельно, воспользуйтесь модулями PAM, например, **pam_cracklib** или **pam_passwdqc**, для принудительной проверки стойкости.

2 Запретим вход как root

Если установить параметр **PermitRootLogin** в файле конфигурации *sshd* в **no**, то вход в систему от имени root будет запрещен. Это усложнит задачу злоумышленника: сначала ему придется угадать имя обычного пользователя и его пароль, а потом – пароль суперпользователя root. Такая защита по умолчанию включена в Ubuntu, хотя и по другой причине – пароль root там заблокирован. Однако это не особо удобно и для клиентов, которым нужно войти в систему как root, потому что им сначала нужно будет сделать это под обычным пользователем, а потом – под root.

3 Уберем подключение с паролем

Хотя SSH поддерживает несколько механизмов аутентификации, по умолчанию обычно активны только два – парольная аутентификация и аутентификация с публичным/приватным ключом RSA. Установив параметр **PasswordAuthentication** в файле конфигурации *sshd* в **no**, вы заставите *sshd* использовать аутентификацию RSA. Перед этим, впрочем, нужно скопировать публичный ключ на сервер. Это нетрудно: если у вас еще нет пары публичный/приватный ключ, запустите **ssh-keygen**, чтобы создать ее, задайте парольную фразу и воспользуйтесь **ssh-copy-id**, чтобы скопировать публичный ключ в файл **authorized_keys** на сервере. Это в самом деле эффективная защита, но если вы регулярно входите в систему с разных компьютеров, надо носить закрытый ключ с собой.

4 Запустим SSH на нетипичном порту

Порт, который слушает *sshd*, можно изменить, установив параметр **Port** в конфигурационном файле сервера. Измените его номер на «случайный» и большой. Конечно, его нужно будет указывать при подключении:

```
$ ssh -p 22550 chris@myserver.example.com
```

Как вариант, можно также настроить это на каждом хосте, в конфигурационном файле клиента SSH (**/etc/ssh/ssh_config**).

Пуристы в информационной безопасности укажут мне, что полной безопасности нельзя добиться, только пустив пыль в глаза. Если вы твердо решите взломать мой компьютер, то без труда определите порт, который слушает *sshd*, с помощью сканера портов *nmap*. Но в большинстве атак на SSH методом перебора используется только порт 22. Могу предположить (хотя это только предположение), что перемещение SSH с порта 22 уменьшит число атак на добрых 90 %. Обеспечит ли это безопасность компьютера? Не абсолютно. Снизит ли риск? Да. Так или иначе, это, видимо, самая слабая мера из здесь приведенных.

5 Гоним нарушителей-рецидивистов

Недавно мне попалась программа *fail2ban*, которая блокирует сайты, неудачные попытки входа в систему с которых повторяются. Ее идея очень проста. Время от времени она заглядывает в журналы, находит сообщения о неудачных попытках входа в систему (с помощью регулярных выражений) и извлекает IP-адрес нарушителя. По превышении определенного порогового значения (его можно задать), *fail2ban* меняет правила брандмауэра *Iptables* таким образом, что IP-адреса нарушителей полностью блокируются. Запрет может быть постоянным или сниматься по истечении определенного времени.

6 Ограничим частоту подключений

Идея здесь состоит в том, чтобы ограничить частоту, с которой можно устанавливать новые SSH-соединения с одного IP-адреса. Если она превышена, поместите IP-адрес в черный список (предполагая, что это злоумышленник) и отбрасывайте (**DROP**) все соединения с данного адреса час или около того. Все это умеет делать новый модуль *Iptables*. Обратите внимание, что данное решение работает только на IP-уровне, и отличить удачный вход в систему от неудачного невозможно – ограничивается только число попыток. Хорошее обсуждение данной технологии имеется на сайте www.sollers.ca/blog/2008/iptables_recent.

› Если вы хотите узнать об SSH больше, материалов очень много..



Wireshark

Wireshark Интересно, что на самом деле происходит в вашей сети? Wireshark поможет это выяснить.

Wireshark – одна из моих любимых утилит. Думаю, потому, что с ней можно делать открытия. Я еще помню то удивление, которое испытал, когда попробовал ее в первый раз (в те времена она называлась *Ethereal*). Наверное, так же чувствовал себя Антоний ван Левенгук, впервые заглянув в микроскоп и увидев то, чего не видел до него ни один человек. Пожалуй, я немного отвлекся, но *Wireshark* – это, несомненно, и удобное средство диагностики, и увлекательный инструмент для изучения того, как все работает.

Wireshark – это утилита, перехватывающая сетевой трафик с одного или нескольких сетевых интерфейсов. Можно задать набор правил, определяющих интересующие вас пакеты. Затем эти пакеты откладываются в сторонку (точнее сказать, в буфер) и всесторонне анализируются. Перехваченные пакеты можно также сохранить в файл и загрузить из него. Обычно *Wireshark* переводит сетевой интерфейс в так называемый «неразборчивый» режим (promiscuous mode) и принимает все пакеты, а не только адресованные вашему компьютеру. Для перевода в «неразборчивый» режим нужны привилегии суперпользователя, поэтому *Wireshark* обычно запускается от имени root.

Начнем с простого примера работы *Wireshark*: рассмотрим результат обращения к web-странице на сервере 192.168.1.67 из браузера на клиенте 192.168.1.69. В верхней части изображения (*выделена зеленым на рисунке внизу*) показан весь обмен пакетами. Каждая строка – это один пакет. Пакеты с первого по третий – открытие TCP-соединения, четвертый – запрос **HTTP GET**, а шестой – ответ на него. Пакеты 7–10 показывают завершение соединения на обоих концах. В колонке Time [Время] показано время, прошедшее с захвата первого пакета в секундах. Оно пригодится, например, для анализа задержек из-за тайм-аутов DNS. В нашем случае на это понадобилось менее 3 мс.

Четвертый пакет на рисунке выделен для подробного анализа. В средней панели мы видим общую информацию о заголовках внутри него для каждого уровня стека протоколов. С помощью маленьких стрелок слева можно раскрыть любой уровень, показав его более подробно. Мы сделали это с заголовком уровня приложения – в данном случае это HTTP-пакет. Теперь мы видим, что это запрос **HTTP GET**, и видим поля заголовка HTTP-запроса.

В нижней части панели отображается содержимое пакета, байт за байтом, в шестнадцатеричном формате и в ASCII. Подсвеченная часть показывает поле заголовка HTTP, выделенное выше – в данном случае, поле **Host:**.

Фильтры

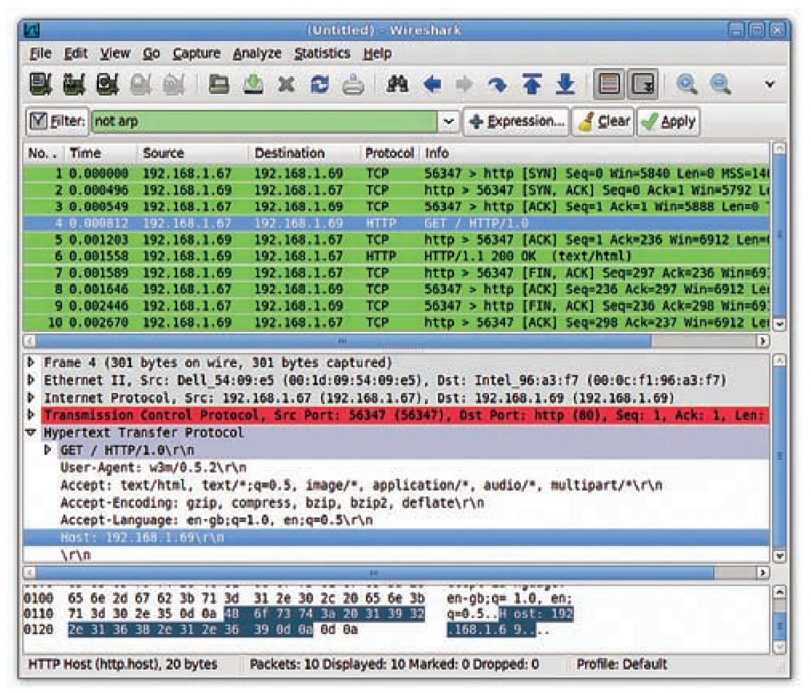
Фильтры – одна из самых мощных возможностей *Wireshark*. Фильтр – это один или несколько тестов содержимого пакета, позволяющих понять, интересен ли он вам. Фильтрация выполняется в две стадии. Фильтры захвата определяют пакеты, которые будут удержаны в буфере захвата, а фильтры отображения определяют, какие пакеты будут показываться. Язык фильтров богат, и фильтровать можно практически по любому полю

Фильтры захвата

Чтобы захватить...	...используйте фильтр
Только трафик, идущий от или к заданному IP	host 192.168.1.44
Только трафик, идущий от или к заданной подсети	net 192.168.1.0/24
Только DNS-трафик (порт 53)	port 53
Все, кроме ARP и DNS	port not 53 and not arp

Фильтры отображения

Чтобы увидеть...	...используйте фильтр
Только трафик между машинами в локальной подсети	ip.src==192.168.0.0/16 and ip.dst==192.168.0.0/16
Только трафик от MAC-адресов устройств Dell	eth.addr[0:3]==00:06:5B
Только HTTP-запросы с URI, заканчивающимися на foo	http.request.uri matches "foo$"
Трафик, имеющий отношение к Windows	smb || nbns || dcerpc || nbss || dns



› **Главное окно *Wireshark*. Все, что вы хотели знать о своей сети, в приятной цветовой гамме.**

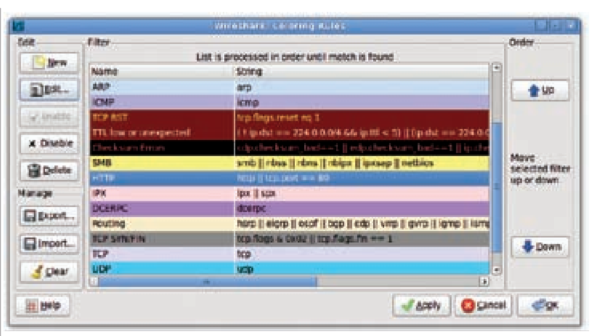
Спросите разрешения!

Если вы хотите запустить Wireshark в своей корпоративной сети или сети провайдера, наверное, стоит сначала спросить разрешения. Хотя это совершенно пассивная и никогда не вторгающаяся утилита (и в большинстве сетей все равно покажет только «ваши» пакеты), корпоративная политика может запрещать применение таких средств. Я вас предупредил!

За кадром

За кадром *Wireshark* всю работу делает программа *Dumpcap*, которая, в свою очередь, использует библиотеку захвата и фильтрации пакетов *Libpcap*, где и находится настоящий движок *Wireshark*.

Эта библиотека используется и другими утилитами перехвата сетевого трафика: например, *Tcpdump*, утилитой сканирования портов *Nmap*, системой обнаружения вторжений *Snort* и *Tshark* – собратом *Wireshark*, работающим с командной строки.



› Цветовое правило связывает выражение фильтра с отображаемым цветом. Цвет каждого пакета определяется первым соответствующим ему правилом.

любого из протоколов, что позволяет сконцентрироваться на трафике, который вам интересен.

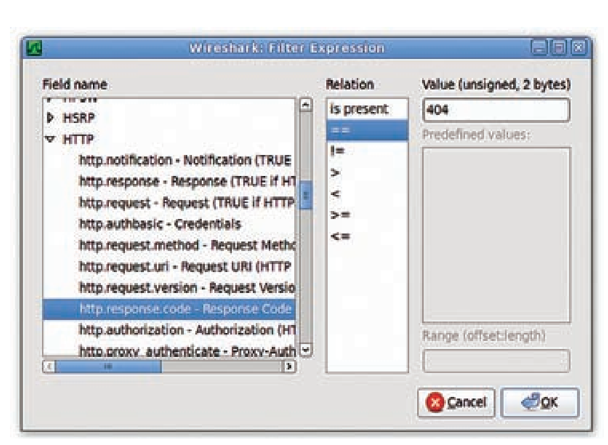
Наблюдательные читатели могли заметить, какой фильтр использовался в последнем примере. Он состоит из простого правила **not arp** и используется, чтобы подавить ARP-болтовню с моего широкополосного маршрутизатора.

Фильтры могут оперировать с заданными протоколами, такими как IP, TCP, UDP, ARP и т.д. Они могут выполнять сравнение на равенство и неравенство и числовое сравнение значений полей. Для полей со строковыми значениями можно анализировать подстроки с помощью оператора **contains** и проверять их на соответствие регулярным выражениям оператором **matches**. Можно даже сравнить содержимое заданной части пакета с помощью синтаксиса **[смещение:длина]**. Отдельные условия можно объединить операторами **and**, **not** и **or**. Все это образует универсальный и мощный язык фильтров.

В *Wireshark* есть графическая утилита, помогающая создавать фильтры отображения. Чтобы запустить ее, нажмите на кнопку Expression [Выражение] на панели инструментов Filter [Фильтр] главного окна. На рисунке *ниже* с его помощью определяется фильтр для отображения только тех пакетов, которые содержат код ответа HTTP 404 («Файл не найден»). Результирующее правило фильтрации: **http.response.code == 404**. Полей, по которым можно фильтровать, полным-полно; а как узнать их имена? Ну, при перемещении по пакету в главном окне *Wireshark* имя поля выбранного элемента отображается в строке состояния. Имена полей можно использовать в выражениях фильтров. Вглядевшись в строку состояния первого экранного снимка, вы увидите, что выбранному на панели выше полю соответствует **http.host**. Полный список имеется на сайте www.wireshark.org/docs/dfref.

В двух таблицах на *предыдущей странице* показано несколько примеров фильтров захвата и отображения, способных дать представление об их возможностях. Прежде чем покончить с фильтрами, скажу пару слов о последнем классе, о котором нужно знать: он позволяет задать цветовые правила для *Wireshark*. Редактор цветовых правил поможет создать новые правила (используя тот же синтаксис, что и у фильтров) и цвета для них, а также импортировать или экспортировать набор цветовых правил. Также можно применять пользовательские цветовые правила для подсветки пакетов на фоне другого трафика. На сайте www.wireshark.org есть набор готовых цветовых правил, с которых можно начать.

А в окне параметров захвата можно задать – не ждите призов за отгадку – параметры захвата. Здесь можно указать интерфейс, пакеты с которого будут захватываться, переключить его в «неразборчивый» режим, задать фильтр захвата и установить предельные значения для числа пакетов, объема данных или времени.



› Редактор выражений фильтров поможет просмотреть доступные поля и построить выражения фильтров .

Делаем это с командной строки

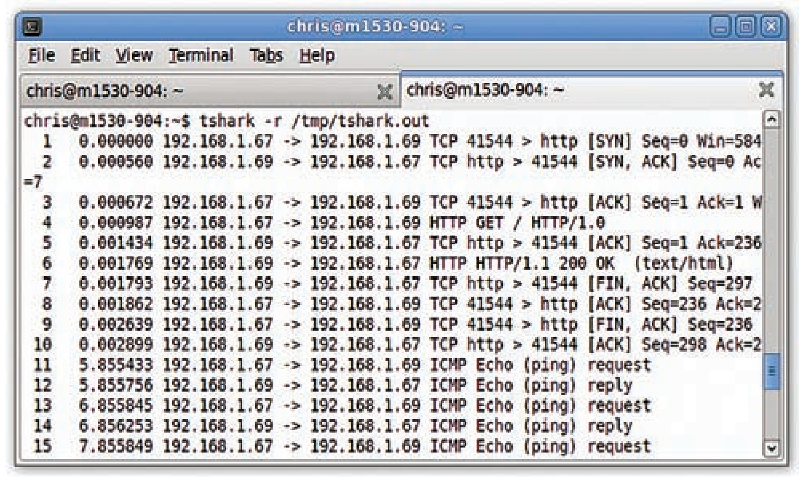
Пакеты можно перехватывать и из командной строки с помощью *Tshark*. Например, опишем перехват ста пакетов с интерфейса **eth0** в файл **/tmp/shark.out**, с фильтрацией ARP-трафика:

```
$ sudo tshark -i eth0 -c 100 -w /tmp/shark.out not arp
```

(Фильтры *Tshark* аналогичны фильтрам *Wireshark*.) Потом можно отобразить трафик, прочитав его командой

```
$ tshark -r /tmp/shark.out
```

На рисунке *ниже* показан фрагмент этого вывода. Тот же файл можно загрузить в *Wireshark*, чтобы просмотреть его в графическом виде. LXF



› У утилиты командной строки *Tshark* те же возможностями по перехвату и фильтрации пакетов, что и у *Wireshark*, но выглядит она далеко не так приятно.

Где узнать больше

Официальное описание синтаксиса фильтров можно найти на man-странице *Tcpdump*. На сайте wiki.wireshark.com также имеется руководство пользователя *Wireshark*, HOWTO, примеры цветовых правил и правил фильтрации, а слишком уж подробное руководство по фильтрам есть на сайте www.wireshark.org/docs/man-pages/wireshark-filter.html.



Linux center
www.linuxcenter.ru

LINUX FORMAT

Учебники

Наши эксперты растолкуют для вас работу с любым приложением Linux!



ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Каждой программе – по своей ОС

Все компании, занимающиеся операционными системами, стремятся предложить конкурентоспособную поддержку доступа в Интернет.
Уильям Генри Гейтс III

Когда вы будете читать эти строки, то нижесказанное для вас уже не будет новостью. Но так как произошедшее событие является довольно знаковым, то нельзя его не прокомментировать. О чем это я? Компания Google заявила, что в ее недрах создается *Google Chrome OS*. Я пока знаю только то, что интерфейсом ко всему будет *Chrome* и что в качестве ядра будет использоваться Linux. Кстати, о последнем говорится несколько вскользь, и какой процент там останется от GNU – не очень понятно (у вас сейчас уже должно быть больше информации). Но в любом случае, очевидно, польза будет всем. Драйверов для Linux тоже должно прибавиться.

Лично меня такое развитие ситуации совершенно не удивило. Linux давно показал, что он является идеальной платформой для создания готового решения, а сервисам Google оказалось нужно таковое. Эти две технологии просто нашли друг друга.

Примерно то же самое уже случилось с домашними и не очень маршрутизаторами, дисковыми сетевыми хранилищами и точками доступа. Теперь пришло время браузеров.

P.S. Добрый человек пообещал за июль «допилить» *Вангеров* под Linux. Успехов ему и безоблачной погоды.

E.m.Baldin@inp.nsk.su

В этом месяце вы научитесь...

Совет месяца: Woof

Обмен файлами через NFS или *Samba* прост, когда они уже настроены на обоих компьютерах, но что если вам нужно по-быстрому перекинуть файл с машины на машину, не связываясь со сложным ПО? Если файл достаточно мал, его можно отправить по электронной почте. Если компьютеры находятся в одной комнате и USB-порты на них не закрыты, поможет флэш-брелок. А есть и еще один интересный метод.

Woof – сценарий Python, который будет работать в любой Linux- (или схожей) системе. Его имя означает Web Offer One File [Web-предложение на один файл], и он представляет собой «одноразовый» web-сервер. Устанавливать ничего не нужно: просто скачайте программу с домашней страницы по адресу www.home.unix-ag.org/simon/woof.html, пометьте сценарий как исполняемый и сделайте требуемый файл доступным для скачивания командой:

```
./woof /путь/к/файлу
```

Вам будет выдан URL, который нужно ввести в браузере на другом компьютере в сети – никакое специальное ПО не требуется. По запросу, *Woof* передаст файл и спокойно завершит свою работу (чтобы разрешить несколько загрузок одного и того же файла, используйте опцию **-c**). *Woof* может обслуживать и целый каталог, например:

```
./woof -z /путь/к/каталогу
```

В данном случае он передаст сжатый *gzip* tar-архив содержимого указанной директории; измените ключ на **-j** или **-u**, чтобы использовать *bzip2* или отключить сжатие. Если вам понравился *Woof* и вы хотите поделиться им с друзьями, раздавайте копии с помощью

```
./woof -s
```

Sockso: Соберем

Устали переписывать музыку с компьютера на компьютер? **Энди Ченнел** создаст вам личный музыкальный сервер, чтобы вы могли слушать свои мелодии отовсюду.





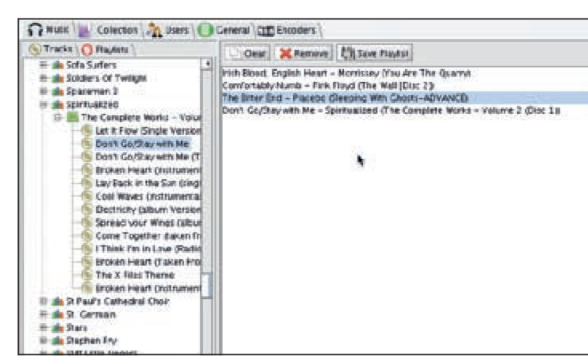
Наш эксперт

Энди Ченнел
Энди, видимо, будет делать Первые шаги в Linux вечно, а технологиями он заинтересовался, открыв для себя Dragon 32.

Мы любим музыку и любим компьютеры. Совместим и то, и другое – и получится идеальная комбинация. Нет, не Kraftwerk, а сервер, способный проигрывать ваши MP3 с любого компьютера. На данном уроке мы скачаем и установим открытый медиа-сервер, настроим его для проигрывания нашей коллекции, создадим плей-листы и обеспечим виртуальный доступ к ним по локальной сети. Мы также рассмотрим возможность доступа к вашей фонотеке за пределами домашней сети.



› Вы можете обращаться к своей музыке через удобный сайт.



› Создайте ваши плей-листы на сервере, перетаскивая треки из дерева файлов в окно списка воспроизведения.

Приложение, которым мы воспользуемся, называется *Sockso*; свежая версия, 11.8 имеется на сайте http://sockso.pu-gh.com. *Sockso* работает под Linux, Mac и Windows и не требует установки как таковой. Вместо этого распакуйте дистрибутив в подходящую директорию и запустите файл **sockso.jar** или скрипт **linux.sh** двойным щелчком. Файл JAR означает, что *Sockso* – это ПО на основе *Java,* и если *Java* у вас пока нет, найдите и установите версию 1.6 *Java Runtime* через ваш менеджер пакетов.

При первом запуске *Sockso* вы увидите главный интерфейс. В нем находятся серии вкладок вверху, сообщение о вашем IP-адресе внизу и двухпанельная секция между ними. В дальнейшем это пригодится для доступа к вашей библиотеке и для создания плей-листов, но сейчас они выглядят чуточку одиноко. Нужно добавить немного музыки.

Эй, диджей, поставь мой диск

Начнем с добавления треков: выберите вкладку Collection [Коллекция] в окне вверху. Появится большой диалог и пара кнопок, нужных для добавления и удаления папок из вашей коллекции. Этот инструмент импортирует подпапки, поэтому не нужно перечислять альбомы по отдельности – просто выберите общий каталог с музыкой, и все подкаталоги будут добавлены на сервер. Ссылки на треки создаются внутри базы данных программы – сами файлы в новое местоположение не копируются.

Мы добавили папку **~/home/user/Music** (здесь **user** – имя администратора сервера), и на внесение в базу данных 3600 песен у нас ушло около трех минут. Возможно, этот процесс займет больше времени, если ваша коллекция крупнее, посему вы можете пропустить эти первые шаги и установить *Sockso* с движком *MySQL*, оптимизированным для работы с большими фонотеками.

По умолчанию *Sockso* просматривает все добавленные папки раз в пять минут, отыскивая новые треки. Если для вас это слишком часто, можете изменить сканирование на ежечасное, перейдя на вкладку General [Общее] и установив Scan Interval [Интервал

» **Месяц назад** Мы организовали работу с *Basket Note Pads* и схемами *Ooo*.



аудиосервер

сканирования] в 60. Если вы добавили треки и хотите сразу обновить коллекцию, перейдите в Collection [Коллекция] и выберите Scan Now [Сканировать сейчас]. А пока вы находитесь во вкладке General, почему бы не дать вашему серверу звучное новое имя – например, Неукротимая Музыкальная Машина LXF?

Теперь неплохо бы протестировать сервер, поэтому запустите свой браузер и наберите http://localhost:4444 в адресной строке. Если все настроено правильно, вы увидите стандартный интерфейс *Sockso*: множество ссылок под ярким заголовком и полоса алфавита с правой стороны окна. Выбор буквы приведет вас к списку исполнителей, имена которых начинаются с нее – справа у каждого имени исполнителя значится количество доступных альбомов, а слева располагаются три иконки. Первая проигрывает все треки этого исполнителя, вторая добавляет треки в плей-лист, а третья скачивает композиции (на нее нажимать не нужно, поскольку вы запустили сервер локально). Щелчок на имени исполнителя выведет список альбомов, а по выбору альбома отображаются отдельные треки.

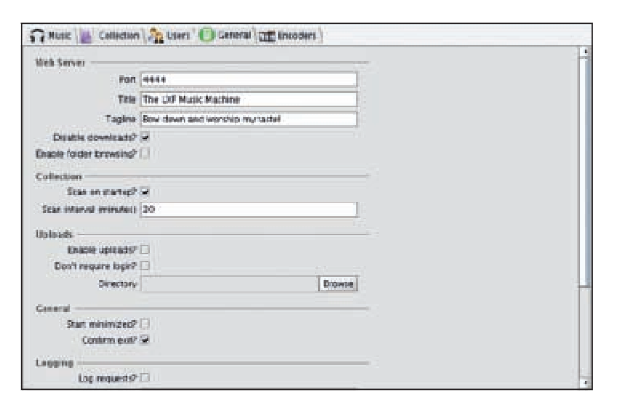
Панель поиска поможет найти определенного исполнителя или песню, в динамике: список доступных треков будет сокращаться по мере набора букв.

Создание плей-листов

Вот бы иметь доступ к плей-листам со всего дома! А с *Sockso* это легко. Перейдите на вкладку Music [Музыка], и слева вы увидите вашу музыку. Для отображения альбомов нажмите на иконку разворачивания (+) рядом с каждым исполнителем, а затем пиктограмму рядом с альбомом для отображения треков. Теперь можно перетащить их в окно справа. Порядок песен можно изменить, перетаскивая их вверх или вниз по списку, а удаление выполняется кнопкой Delete.

Получив подходящий вариант списка, нажмите Save Playlist [Сохранить плей-лист] и введите название. Для просмотра плей-листов, нажмите на вкладку Playlist [Плей-лист] вверху дерева файлов. Для редактирования одного их них, щелкните и перетащите его в правую панель. Внеся изменения, нажмите Save Playlist снова и введите название нового плей-листа.

Вернитесь к серверу и нажмите на ссылку Playlist [Плей-лист]. Созданные вами плей-листы будут отображены в разделе Site Playlists [Плей-листы на сайте], а справа вы увидите User Playlists [Плей-листы пользователей], которые позволяют другим пользователям создавать собственные плей-листы через web-интерфейс.



› **Вы можете запретить скачивание, чтобы копии ваших композиций не сохранялись повсеместно.**

Их плей-листы будут отображаться в интерфейсе администрирования во вкладке Playlists, и если некий пользователь создаст замечательный плей-лист, вы можете скопировать его себе. Просто перетащите его в окно Playlist (справа), сделайте изменения и затем сохраните его под новым именем.

Доступ по сети

Теперь представьте, что в вашем доме четыре пользователя, желающие иметь доступ к серверу. Правду сказать, каждый человек может вводить локальный IP-адрес сервера в браузере, но он скорее всего будет не тем, что указан в *Sockso*. Вы можете выяснить адрес компьютера в терминале, набрав **ifconfig.** Введите его в браузере, добавив **:4444** – порт нашего сервера.

Позаботимся, чтобы каждый из наших пользователей регистрировался для доступа к музыке, и ограничим скачивание. Для этого перейдите во вкладку Users [Пользователи] и нажмите кнопку Create User [Новый пользователь]. Укажите имя пользователя, пароль (дважды) и адрес электронной почты, затем нажмите на Create User. Перед выходом из этого раздела нажмите на кнопку Require Log In [Требовать регистрации]. Завершите процесс, вернувшись во вкладку General и выбрав Disable Downloads [Запретить скачивание]. Ваша домашняя радиостанция готова. Блестяще!



Если вы склонны покопаться в коде, то вся информация по архитектуре *Sockso* найдется в папке **application/resources.**

› **Всплывающий плейер функционален без примитива.**

Воспроизведение

По умолчанию все плей-листы и файлы, выбранные для проигрывания, открываются во всплывающем *Flash*-плейере. Это работало на всех машинах, которые мы настраивали, независимо от ОС, а также позволяет запустить длинный плей-лист и затем закрыть web-интерфейс *Sockso*, не теряя музыки. Однако если у вас возникают проблемы с *Flash*, есть и альтернативные возможности: например, передача на другие приложения, типа *iTunes* или *Windows Media Player*, или использование встроенного плейера.

Flash отображает плей-лист для выбора треков для воспроизведения, набор инструментов навигации и обложку альбома, трек из которого играет в данный момент.

Basket: Порядок

Basket не только хранит списки задач – это идеальный инструмент исследователя.

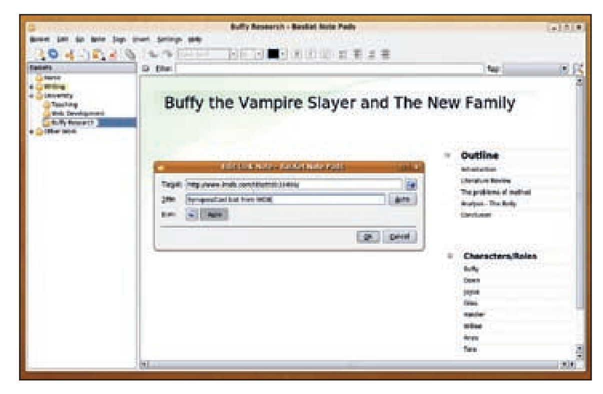
В прошлом месяце мы создали базовую систему *Basket* для работы со списками задач и простыми заметками. Теперь рассмотрим другие способы применения этого мощного приложения для организации, управления и совместного использования информации. А именно, применим его как утилиту для исследования, полезную при написании романа, приготовлении доклада, создании презентации или планировании дел по дому. Мы также рассмотрим создание, редактирование и организацию заметок, импорт информации из различных источников, совместное использование сведений с друзьями и коллегами и применение *Basket Note Pads* для представления вашего исследования другим.

Подготовим приложение

Сначала создайте новую корзину: это делается щелчком правой кнопки мыши на списке Корзин [Baskets] или нажатием Ctrl+N. **Модифицируйте ее с помощью различных опций (мы ос**тановились на корзине свободной формы), укажите имя и нажмите OK. Пространство справа станет активной корзиной, настраиваемой щелчком на названии правой кнопкой и выбором Properties [Свойства]. Мы можем дать корзине имя, щелкнув правой кнопкой в любом месте и выбрав пункт Text [Текст]. Появится поле, где можно набрать текст и отредактировать его, используя инструменты вверху окна.

Также можно добавить списки (см. **LFX120** для более подробной информации), сгруппировать их вместе для облегчения управлением корзиной и создать дополнительные текстовые элементы. Собрав в окне несколько объектов, неплохо будет попрактиковаться в их перемещении: то, что передвигается, зависит от того, где вы взялись перемещать элемент. Например, если вы сгруппировали несколько элементов вместе, можно удалить объект из группы, наведя курсор мыши рядом с его именем (голубое выделение покроет только этот элемент) и перетащив его. Таким же способом можно менять порядок элементов в стеке. Передвиньте мышь немного влево, и выделение должно покрыть всю группу, которую также можно переместить.

Следующая вещь, которую мы хотим добавить – изображение; это можно сделать, перетащив его файл из любой папки в активную корзину При этом вам будет задан вопрос, хотите ли вы скопировать или переместить изображение. Перемещение удалит изображение из исходного местоположения, и вы скорее всего предпочтете копирование. Изображение, попавшее в корзину, можно перемещать, как и другие элементы. Мы дали картинке имя, а затем сгруппировали два элемента вместе – это делается либо нажатием сначала на первом, а затем – с клавишей Shift – на втором, плюс щелчок по иконке скрепки в панели инструментов, либо перетаскиванием одного из элементов так, чтобы он прикрепился к другому (вы увидите появляющуюся толстую черную линию под исходным объектом). С помощью второго метода также возможно, например, убрать изображение под его заголовок, нажав на иконку со знаком минус. Таким образом можно создать сложные блоки упорядоченной информации и затем одним щелчком скрыть их за названиями.



› **Умение *Basket* работать с web-контентом и другими приложениями делает его полезным инструментом для исследований.**



Сохраните ваш файл Basket в папке *Dropbox* (www.getdropbox.com), чтобы иметь доступ к нему с любого компьютера.

› **Успешное создание структур в *OOo* зависит от стилей абзацев и Навигатора.**

Визуально

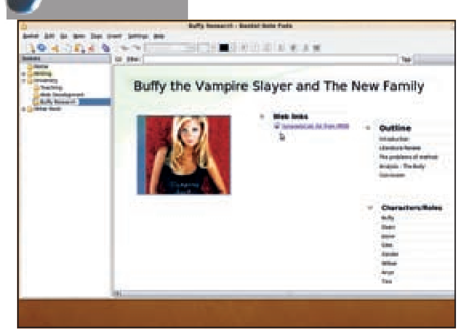
При желании исправить изображение, сделайте на нем двойной щелчок, и оно откроется в соответствующем редакторе. После завершения, ваши изменения автоматически сохранятся в версии *Baskets*, благодаря чему не нужно импортировать изображение дважды.

Можно также создать на ваше изображение ассоциативные ссылки (или куски текста): сначала выберите его (выделите, поводив около него курсором, пока не появится граница, и нажмите на нем), затем перейдите в Insert > Link [Вставить > Ссылка] и введите необходимую информацию. Ссылка дополнит исходный текст или изображение, а при нажатии на иконку минуса оно, как и раньше, свернется. В нашей исследовательской корзине, мы создали раздел под названием «Web-ссылки» для размещения коллекции ссылок, связанных с проектом.

Однако тщательное исследование основывается на сборе информации из различных источников, поэтому просмотрите пошаговое руководство для более детальной информации об интеграции *Baskets* с другими Linux приложениями и *Firefox.*

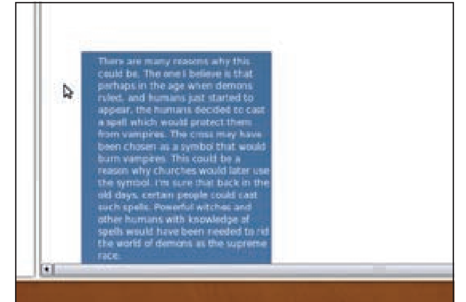
в исследованиях

Шаг за шагом: Создание исследования



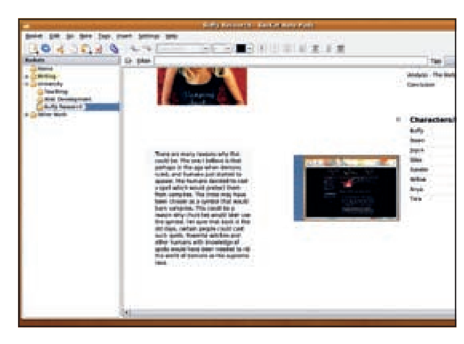
1 Перенос ссылки

Для архивирования полезных ссылок, найденных в Интернете, перейдите по адресу, который вы хотите сохранить, выберите маленькую иконку [favicon] в адресной строке и перетащите ее в свою корзину. Ссылка будет отформатирована с названием страницы, и ее можно перемещать, как и другие элементы *Basket Note Pads*.



2 Взять текст в Интернете

Найдя интересный кусок текста в Интернете, выделите его и перетащите из вашего web-браузера (или приложения) в свою корзину, где он появится как новый элемент. Учтите, такой текст не сохраняет информацию об источнике, и, возможно, полезно будет добавить ссылку на последний. Этот текст можно редактировать обычным способом.



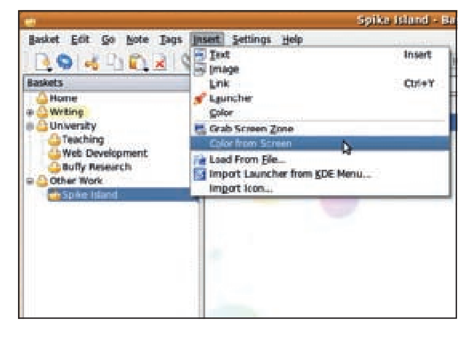
3 Сделайте снимок экрана

Проблема ссылок заключается в том, что они со временем устаревают, или текст может измениться. Решить ее можно, сделав снимок экрана. Щелкните правой кнопкой в любом месте корзины и выберите Grab Screen Zone [Захватить участок экрана]. *Basket* свернется, а курсор превратится в крестик. Выберите кусок, который хотите сохранить, и он будет вставлен в корзину.



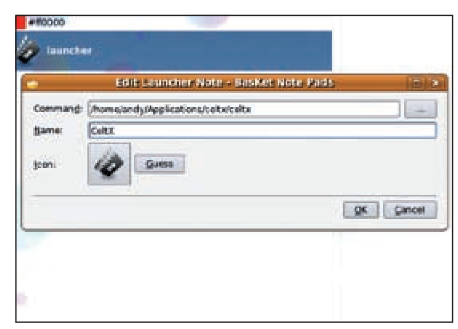
4 Экспорт HTML-страницы

HTML – отличный способ сделать резервную копию корзины или разрешить к ней совместный доступ друзьям или коллегам, не владеющим *Basket*. Перейдите в File > Export > HTML Page [Файл > Экспорт > HTML-страница] и выберите местоположение файла. Создастся отдельный HTML-файл и папка со связанными с ним элементами. Ссылки в корзине остаются активными, делая *Basket* хорошим, хотя и простеньким, web-редактором.



5 Создание палитры цветов

Если вы используете это приложение для управления дизайном проекта, можно создать палитру цветов с экрана. Например, откройте логотип на вашем рабочем столе в *GIMP*, выберите Insert > Colour From Screen [Вставка > Цвет с экрана] и используйте пипетку для выбора цвета. В результате вы получите образчик оттенка вместе с его шестнадцатеричным кодом.



6 Создайте кнопку запуска

Наконец, с исследованием можно связать приложения – например, *CeltXP* пригодится при работе над сценарием. Для этого щелкните правой кнопкой и выберите Launch [Запуск]. Укажите ваше приложение в списке или перейдите к нему, используя иконку (...). Теперь нажмите OK – создастся кнопка запуска вашего приложения. LXF

GIMP: Создаем

Майкл Дж. Хэммел в этом месяце продолжает полет на хрупких крыльях вдохновения. Запускайте *GIMP* и присоединяйтесь к его путешествию!



Берем из стока

- **Основное изображение**
  Девушка с развевающимися волосами: bigstockphoto.com/photo/view/1921777
- **Декоративные рамы с орнаментом**
  Позолоченная рама для картины: istockphoto.com/stock-photo-1710947-golden-picture-frame.php
  Старинная позолоченная рама: istockphoto.com/stock-photo-4930364-vintage-gold-frame.php
- **Пламя**
  Были использованы фотографии:
  sxc.hu/browse.phtml?f=view&id=1093986
  sxc.hu/browse.phtml?f=view&id=1099709
  sxc.hu/browse.phtml?f=view&id=1104673
  sxc.hu/browse.phtml?f=view&id=1143736



Наш эксперт

Майкл Дж. Хэммел
Участник разработки *GIMP* и автор трех книг по данной тематике, включая самую новую – «Эффекты в *GIMP*: руководство для художников».

После почти тридцати лет создания ПО и работы с ним мне иногда бывает трудно найти тему для проекта, которая бы меня захватила. То же относится к моим графическим произведениям; и, наткнувшись на нечто вдохновляющее, я набрасываюсь на это, как какой-нибудь исполнительный директор на фонд финансовой помощи. Вдохновение – это валюта, и тратить ее надо быстро, не то курс упадет.

Поэтому прошу у моих дорогих читателей прощения: мой энтузиазм разгорелся под влиянием еще одного проекта профессионального уровня. Как и в прошлом месяце, этот учебник скорее нацелит подающих надежды в *GIMP* на поиск и комбинирование идей, чем откроет секреты реализации какой-то конкретной задачи, хотя сам процесс я тоже освещу.

Возродим отгоревшее творчество

Эта работа, которую я назвал «Богиня Огня», основана на двух интернет-руководствах, которые я объединил в одно. Первая вдохновившая меня статья, Mystic (http://psdtuts.com/designingtutorials/the-making-of-mystic), описывает создание в *Photoshop* того, что я могу приблизительно описать как божество древних майя. Процесс полностью воспроизводится в *GIMP*, но я не сумел найти подходящие фрагменты с орнаментом.

Второй источник моего вдохновения – другое руководство для *Photoshop*, где создается изображение женщины, объятой пламенем: http://psd.tutsplus.com/tutorials/tutorials-effects/how-to-create-a-flaming-photo-manipulation. Эту работу также можно воспроизвести в *GIMP*, но отсутствие аналога инструмента Transform осложняет процесс, хотя и не делает его невозможным.

Идеи обоих статей дополняют друг друга и зовут к экспериментам. При желании в точности повторить слияние этих двух работ, обзаведитесь стоковыми изображениями – их можно получить бесплатно или по скромной цене из фотобанков (см. врезку вверху «Берем из стока»). Пакет *Gimp Paint Shop*, который мы использовали на уроках последние пару месяцев, здесь не обязателен, но его присутствие предоставит вам больше удобства.

Я создал изображение размером 2350 × 2033, но если надо сократить требования к памяти и ресурсам процессора – прежде чем приступать, уменьшите масштаб стоковых изображений вдвое.



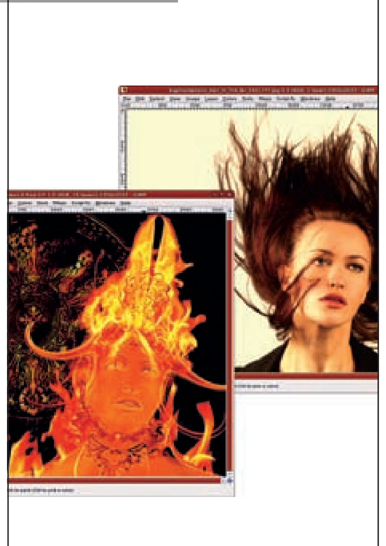
› **Результат ваших тяжких усилий — вот эта огненная красотка.**

» **Месяц назад** *GIMP* в союзе с графическим планшетом разрушил город.

богиню огня



Шаг за шагом: Гори весь мир огнем!



1 Простые действия

Сравним изображение-оригинал и законченную работу. Отличий у них хватает, но главных перемен две: во-первых, я удалил волосы девушки, а во-вторых, окрасил изображение в цвета огня.

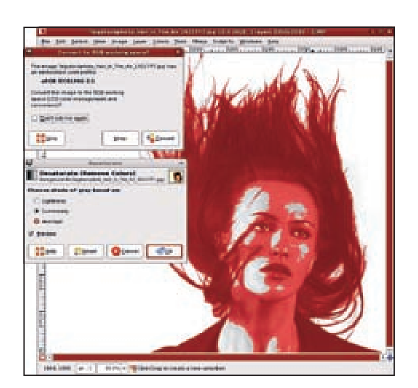
Богиня лишена волос, потому что сперва я планировал как можно ближе следовать руководству по созданию пылающей девушки и, соответственно, выбрал базовое изображение с волосами, которые в *GIMP* легко было бы превратить в пламя. Но поскольку в этом процессе применялась функция *Photoshop*, которую довольно трудно воспроизвести в *GIMP* (а именно, Warp Transform), то я изменил проект, и волосы стали не нужны.

По мере продвижения вы заметите, что пламя не только является компонентом изображения, но также используется для окраски. Простой способ окрасить слой огнем – это убрать насыщенность слоя, подлежащего окраске, расположить над ним слой с огнем и выставить для него подходящий режим. Но если вы расположите что-либо под этим слоем, черная область вокруг огня сделает изображение невидимым. Чтобы этого избежать, применим для слоев с огнем маски или же просто скопируем и вставим языки пламени (без черного фона) из исходных изображений.

В итоге процесс создания Богини Огня разбивается на несколько основных этапов:

1 Отделение девушки от фона.
2 Определение границ головы и создание нескольких соответствующих слоев для получения точного выделения.
3 Добавление украшений.
4 Удаление более тонких лишних деталей.
5 Окраска изображения огнем.
6 Создание огненной короны.
7 Добавление фоновой эмблемы.

Имейте в виду, что даже при буквальном следовании уроку точно такое же изображение, как у меня, у вас не получится. Перечисленные шаги дают массу вариаций, особенно при создании огненной короны. И не огорчайтесь, если первые попытки будут далеки от идеала. Я сделал семь различных вариантов, прежде чем набить руку в этом процессе, и первые четыре или пять, честно говоря, я бы не показал никому.



2 Отделяем девушку

Первый шаг в этой работе – отделить девушку от фона, в основном потому, что фон из оригинального изображения девушки нам не нужен, и его цвет может породить проблемы при наложении других слоев. Удаление фона также позволяет добавить элементы, поверх которых расположится голова девушки.

Чтобы отделить девушку, сначала обесцветим (Цвет > Обесцветить) исходное изображение с отмеченной опцией Светимость. Причиной выбора этой настройки послужило то, что изображение-оригинал после конвертации цветового пространства (в исходном формате, в каком оно было загружено со стокового сайта) имеет в *GIMP* желтоватый оттенок на мониторе с управлением цветом. Я говорю «с управлением цветом», потому что для моих мониторов Acer X203W я создал профиль в Настройках *GIMP*. К счастью, при выбранной опции Светимость в диалоговом окне обесцвечивания желтый оттенок становится ярко-белым.

Теперь выберем на панели инструментов Выделение по цвету и щелкнем по белому фону. Настроив в параметрах инструмента порог на 50, мы получим полностью выделенный белый фон, немного волос и ряд областей лица и шеи. Чтобы избежать выделения участков кожи, переключимся в Быструю маску, закрасим белые области на девушке большой черной кистью, выйдем из режима Быстрой маски и инвертируем выделение. Это выделение затем скопируем и вставим в новый слой, под названием «Девушка». Затем подгоним слой к холсту с помощью пункта меню Слой > Слой к размеру изображения, чтобы дальнейшие изменения не ограничивались размером слоя.



3 Обстригаем волосы

В исходном слое (который по умолчанию называется «Фон»), я с помощью Ctrl+A выделил весь слой целиком, а затем перетащил черный цвет фона на холст, чтобы полностью заполнить слой черным.

Пора избавить девушку от волос. Как и в большинстве операций, включающих манипуляции с частями изображения, я на самом деле не хочу удалять волосы из самого слоя – лучше сохранить их, прикрыв маской, потому, что позже, возможно, мне захочется опять включить эту часть (в нашем случае, волосы) в изображение. Эта гибкость – большое преимущество цифрового редактирования изображений, и я очень советую не пренебрегать им. В данном случае я добавляю к изображению белую маску слоя. Начав с большой кисти и по мере продвижения работы переключаясь на меньшие размеры, я крашу черным поверх волос девушки до тех пор, пока бо́льшая часть волос не будет удалена. Я оставляю часть волос незакрашенными, потому что собираюсь потом использовать фильтры определения краев и знаю, что после их применения линии в волосах дадут еще одну визуальную черту. После этого я отключаю видимость слоя с девушкой, чтобы он не отвлекал меня на следующих этапах работы. »

4 Определяем границы

Следующий шаг проекта – создание детального контура девушки с помощью двух фильтров выделения краев: Край (Фильтры > Выделение края > Край) и Неон (Фильтры > Выделение края > Неон). Они выдают практически аналогичные результаты, и позже я использую их в сочетании с режимами слоев.

Для начала я дублирую слой с девушкой и называю копию «Края». Затем открываю фильтры выделения края и выбираю алгоритм По Прюитту. Я делаю это на основании предпросмотра: Прюитт обеспечил самые подробные детали. Примените фильтр к слою, затем установите режим слоя на «Экран».



Дублируем слой с девушкой снова; на этот раз назовем копию «Неон». Передвинем ее на вершину стопки слоев и инвертируем (Цвет > Инвертировать), а затем применим фильтр выделения края «Неон». Выставим режим слоя «Экран» и, наконец, скопируем этот слой, что в итоге даст нам три слоя, предназначенные для выделения края и расположенные над слоем девушки. Увеличиваем масштаб и видим пряди волос, которые не удалось замаскировать. Их можно оставить – или убрать с помощью ластика. Во втором случае ластик, возможно, придется использовать на одном и том же участке каждого из трех слоев с выделенным краем. Их трудно разглядеть на печати, поэтому я их тут не показываю. В своей работе, тем не менее, я стер лишние пряди волос – я скрупулезен в работе с деталями.

5 Введем украшения

Теперь принарядим нашу богиню. Лучше всего поискать необходимые орнаменты среди архитектурных украшений или рам для картин. В данном проекте используются только рамы: хороших фото викторианской архитектуры мне найти не удалось.

Я добавлю украшения на лоб и на место соединения головы и шеи, а из еще одного орнамента получится ожерелье. Первые два элемента взяты из прямоугольной рамы, а третий – из овальной.

Начнем с прямоугольной рамы; обесцветим ее. Рама целиком располагается на однотонном фоне, и содержит внутри однотонную область, поэтому выделить ее проще всего будет инструментом Выделение связанной области, с порогом 50. Один щелчок снаружи рамы и один – внутри дадут нам выделение фона. Увеличьте выделение на 1 пиксель (я почти всегда так делаю, чтобы не оставлять более светлых точек вдоль границы выделенного объекта) и инвертируйте выделение. Скопируйте и вставьте раму в рабочий холст как новый слой поверх стопки слоев. Назовем его «Налобное украшение» и убедимся, что режим слоя установлен на «Нормальный».

Сделав это, масштабируем и повернем слой с налобным украшением над лицом девушки. Используем маску слоя, чтобы удалить правый и левый края рамы, а также нижнюю часть. Скопируем слой и выставим режим слоя-оригинала на Осветление. Осветление копии слоя над обесцвеченным слоем в нормальном режиме увеличивает контраст (что придаст украшению заметно больше блеска).



6 Наряжаем богиню дальше

Нашейное украшение получается таким же способом, только с новой копией рамы. Она нужна потому, что угол поворота, размер и положение будут отличаться от версии для лба, не говоря уже о том, что и маска будет другая. В этом слое мы сохраним нижнюю часть рамы, а остальные – замаскируем.

Точно так же получим ожерелье из элементов овальной рамы. На сей раз, прежде чем повернуть раму и изменить ее размер и положение, отразим ее зеркально по вертикали. Наконец, сделаем маску, оставив видимой только нижнюю часть рамы. Заметьте, что изображения рам были гораздо меньше, чем главное окно изображения и девушки. Их нужно увеличить. Увеличение масштаба чревато проблемой: при этом изображение скоро пойдет квадратами. Но в нашем случае этот эффект почти не заметен, благодаря сливающимся контурам частей украшения. С помощью режимов растворения слоев также можно частично скрыть эффект пикселизации, а наш метод окраски спрячет его еще лучше. Однако на больших отпечатках эффект разложения на пиксели скорее всего будет виден. Поэтому, если вы собрались печатать копии изображения, не делайте их больше страницы журнала. (Удобная мерка, правда?)

7 Удаление лишних деталей

Украшения закрывают не столь обширную область волос девушки, как я ожидал, и чтобы избавить нашу богиню от этих неприглядных дефектов, обратимся к инструментам Ластик и Палец.

Можно применить Палец со слоями, определяющими край (белый и черный цвета вместе дают оттенки серого), но я скорее бы оставил некоторые линии на лице. Для этого используем Ластик с кистью Grunge и выставляем динамику кисти на увеличение непрозрачности, жесткости и размера при надавливании на мой планшет Wacom. Стоит заметить, что пока у меня не было нужды прибегать к планшету, но на данном этапе он оказался очень кстати. Если у вас нет графического планшета, просто используйте кисть меньшего размера и изменяйте процент непрозрачности вручную.

Теперь применим ластик по очереди к каждому из слоев, определяющих край. Я слегка поправляю один слой, затем перехожу к следующему, а затем снова повторяю процедуру с каждым слоем. Цель – оставить как можно больше деталей, удаляя заметные линии прядей волос.



8 Смажем остатки волос

Переход к удалению волос означает, что слой «Девушка» снова надо делать видимым. Отключим видимость всех трех слоев, определяющих край, и включим видимость слоя с девушкой. Отметим галочкой опцию Запереть для слоя с девушкой и применим Палец с любой подходящей кистью, чтобы убрать остатки волос из слоя. Постарайтесь выдержать мягкое освещение по всей поверхности щеки, щелкая сначала по более светлым поверхностям (щека), а затем протаскивая к более темным областям (волосы). Подобным образом обрабатываем шею и одежду.

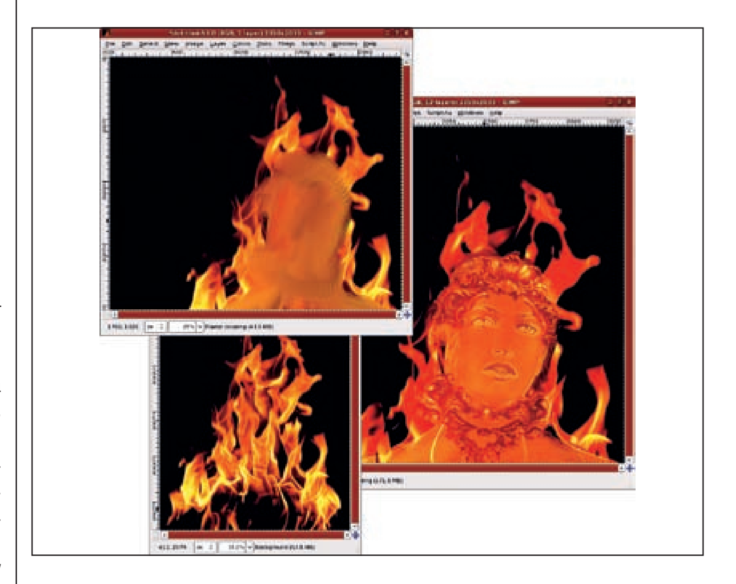
При включенной опции Запереть нечего беспокоиться, что мы «заедем» за границу контура девушки. Хотя в данной работе я того и добивался, вы можете дать волю творчеству, отключив опцию Запереть для слоя и работая по всей поверхности. Не забудьте только позже окрасить эти области огнем. Закончив, снова сделаем слои контуров видимыми.



9 Красим в цвет пламени

Раскрашивание – следующий шаг. Сначала окрасим девушку, просто применяя функцию Тонировать со значениями тона, насыщенности и освещенности, установленными на 30, 90 и -30 соответственно. Далее скопируем изображение пылающего пламени (огонь и черный фон) и добавим как новый слой поверх стопки слоев. Он масштабирован так, чтобы пламя полностью заслонило девушку, а затем обрезан по размеру изображения (Слой > Слой к размеру изображения). Установим режим слоя на направленный свет (Hard Light). В результате кадрирования слой может не полностью покрыть ширину изображения, и будет видна белая полоса. Тогда выберите Выделение связанной области и щелкните по полосе, увеличьте на 1 пиксель и залейте выделение черным цветом.

Сейчас языки пламени скрывают лицо девушки, и у нас есть две возможности: оставить их или применить к слою с языками пламени Палец, чтобы лицо и орнамент проглядывало сквозь них. Я выбрал второй вариант.



10 Создаем огненную корону

Вот мы и добрались до по-настоящему интересной, творческой части. Процесс создания огненной короны необычайно прост, но чтобы получить правильный результат, скорее всего, понадобится не одна попытка. Начать лучше с изображения, подобного показанному внизу – просто огонь, а не пылающее пламя. Скопируем один огонь (без фона) в рабочее окно и назовем новый слой «Огненная корона». Масштабируем его так, чтобы он укладывался вдоль лба девушки, и растянем до верхней границы изображения.

Скопируем слой, зеркально отразим его по горизонтали и поместим рядом с исходным изображением. При необходимости используйте маску, чтобы собрать языки пламени вместе. Затем объединим оба слоя, выделив верхний слой и выбрав в меню Слой > Объединить с предыдущим. Передвинем объединенный слой к середине окна изображения и приведем его размер к размеру изображения (Слой > Слой к размеру изображения). Теперь основа для короны пристроена, и вокруг есть место для создания вихрей пламени.



Для создания вихрей пламени прибегнем к фильтру интерактивного искажения (iWarp). Здесь начинаются сложности. Вихри на левой стороне короны сделаем при помощи режима Swirl CW [вихрь по часовой стрелке], а на правой – при помощи режима Swirl CCW [против]. Для данного конкретного пламени я выбрал два боковых всполоха – из них выйдут отличные вихри. Щелчками (но не протаскиванием) по разным точкам в окне предварительного просмотра я смог сделать два почти одинаковых завихрения. Щелчки и удерживания тоже подходят, но удерживать мышь нужно недолго, иначе завиток будет чересчур велик.

Стоит также перепробовать разные настройки Радиуса и Степени деформации и точки выполнения щелчков в окне предпросмотра. Если предпросмотр вас разочаровал, нажмите на кнопку Reset [Сбросить] и попробуйте снова. Потребуется терпение, но в итоге вы получите приемлемый результат.

Применив настройки искажения к слою пламени, расположим и повернем нужным образом корону над головой девушки. Выставим режим слоя на «Экран» и добавим над ним еще одну копию пылающего пламени, изменив размеры так, чтобы оно целиком уместилось внутри короны. Установим режим слоя для пылающего пламени в «Экран» и с помощью маски слоя мягко добавим его к короне.

11 Рисуем фоновую эмблему

Последний шаг крайне прост. Чтобы привнести в работу немного колорита племени майя, я выбрал кисть Mandala [Нимб] из набора *Gimp Paint Shop* и увеличивал ее размер до тех пор, пока нимб не закрыл примерно три четверти окна изображения. Тогда я добавил прозрачный слой над фоном, сделал цвет переднего плана белым и один раз щелкнул по этому новому слою.



Над нимбом в новом слое я опять поместил изображение пылающего пламени. Изменив размер так, чтобы он покрыл большую часть отпечатка, я выставил режим слоя на Умножение. Вот и все.

Заключение

Ранее я упоминал о том, что подумывал удалить девушку из изображения. Это не давало особого визуального эффекта, пока я не добавил слой с нимбом и не окрасил его пламенем. Теперь после удаления слоя с девушкой, нимб стал виден насквозь и смотрится очень хорошо. Почему бы не взять это за основу для дальнейших экспериментов? Скоро у вас получится дивная вариация на эту тему. LXF



» **Через месяц** Зажжем кристаллы дилития и имитируем быстрое движение.

OOo: Работаем

Марко Фиоретти автоматически создает уйму счетов и тестовых заданий с вариантами ответов в формате OpenDocument, избегая рутинной работы.



Наш эксперт

Марко Фиоретти автор книги The Family Guide to Digital Freedom, а к тому же активист свободного ПО и программист.

Мы часто применяем компьютер для создания нескольких разных версий одного и того же документа, но ручные обновления ваших файлов имеют смысл, только если они редки. Ведь имеется множество способов сэкономить драгоценные секунды – как мы показали в LXF119 при рассмотрении автоматического обновления электронных таблиц. А на данном уроке мы займемся более объемной задачей: перенесем набор «сырых» данных в какой-либо полезный документ с минимумом усилий. Вдобавок мы избежим открытия *OpenOffice.org* (ну, почти), благодаря свойствам его стандартного формата файлов: OpenDocument, или ODF.

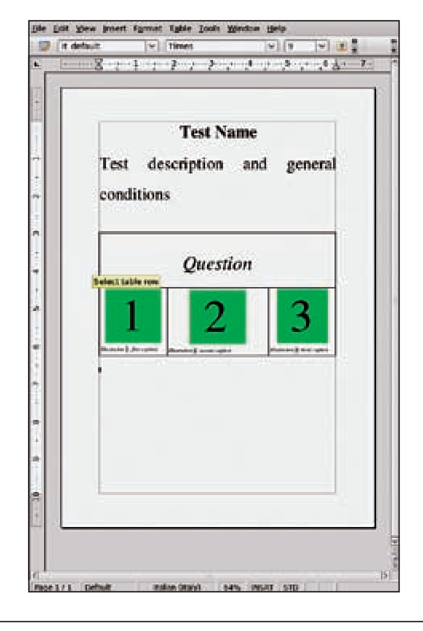
Документ ODF – просто zip-архив, с картинками и макросами в собственных каталогах, а сам текст – записанный в формате XML – находится в файле с именем **content.xml**. Для создания новой версии документа следует открыть архив, изменить текст в **content.xml**, поместить несколько новых картинок в их каталог и запаковать вновь. Здесь мы покажем, как выполнить это с помощью текстовых документов ODF и действительно простых сценариев оболочки.

Тесты с выбором ответа в ODF

Давайте начнем с практической задачи: создания форматированного теста с выбором из нескольких вариантов ответа и случайно генерируемыми вопросами. Ради простоты предположим, что тест содержит всего один вопрос и представляет собой документ, содержащий название теста, его описание, вопрос и три варианта ответа, каждый со своей картинкой и подписью. Его файл (дадим ему имя **sample_multiple_choice.odt**) будет нашим исходным шаблоном. Чтобы использовать файл для генерации нашего теста в формате OpenDocument, сначала надо записать все строки, которое войдут в тест, в файле формата ASCII (назовем его **my_test_data_1**) следующим образом:

```
marco => cat my_test_data_1
TEST_NAME='Тест на совместимость с GNU/Linux'
DESCRIPTION='Это весьма научный тест, выявляющий скрытые оттенки вашего отношения к свободному ПО'
QUESTION_NAME='Какой дистрибутив GNU/Linux вы предпочитаете?'
FIRST_CAPTION='Это Ubuntu?'
SECOND_CAPTION='...или Mandriva?'
THIRD_CAPTION="А может, вы поклонник Fedor'ы?"
```

Формат данного файла достаточно прост, но есть пара моментов, которые следует отметить: во-первых, это вполне корректный синтаксис команд оболочки. Каждая строка этого файла присваивает значение некоторой переменной среды, которая будет использоваться в основном скрипте. Во-вторых, хотя общих правил нет, будьте аккуратны в использовании кавычек. Например, в последней строке нужно использовать двойные кавычки, поскольку в тексте вопроса уже встречается одинарная [или ее необходимо экранировать, заменив на \', – *прим. ред.*].

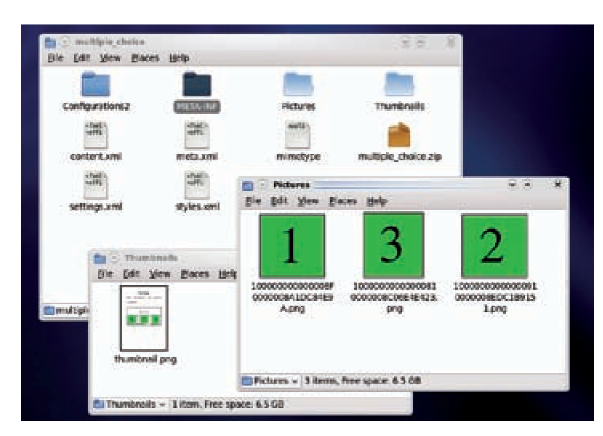


› Благодаря ODF можно создавать документы и без *OpenOffice.org*, но *OOo* все же нужен для написания исходного шаблона.

» **Месяц назад** Мы научились генерировать автоматические слайд-шоу.

автоматом





› В файлах OpenDocument секретов нет: когда вы их распакуете, текст, метаданные и картинки будут сразу видны.

Подготовка картинок

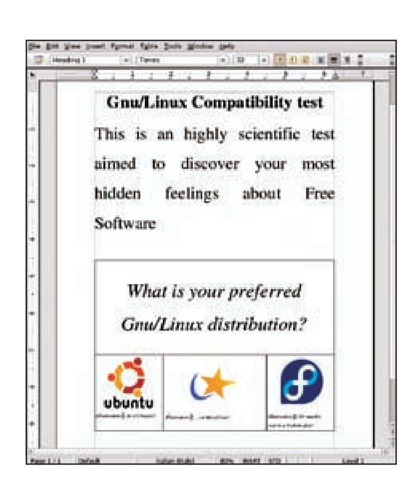
Кроме строк, необходимо также припасти три картинки – в том порядке, в каком они появятся в документе. В нашем примере они находятся в каталоге **Linux_Test_Pictures**, и мы назвали их **01_ubuntu_logo.png**, **02_mandriva_logo.png** и **03_fedora_logo.png** соответственно. Нумерация необходима для гарантии их использования в правильном порядке. Пристроив все по местам, выполните следующую команду:

```
marco => test_generator.sh sample_multiple_choice.odt my_test_data_1 Linux_Test_Pictures/*
```

для завершения вашего первого теста.

Употребим скрипт иначе

Лучшее в этом процессе то, что метод применим для всех случаев, пока структура документа неизменна. В качестве доказательства, разместим три соответствующие картинки (первая – горы, вторая – пляж, третья – исторические места) в каталоге **Holiday_Pictures** и добавим следующий ASCII-текст в файл с именем **my_holiday_data_1**:



› Вот результат перезапуска нашего простого скрипта обработки OpenDocument: первоначальный шаблон заполнен динамическими данными, и все это без *OpenOffice.org*.

```
marco => cat my_holyday_data_1
TEST_NAME='Тест на предпочтения в отдыхе'
DESCRIPTION='Выясняет, как вам лучше провести отпуск'
QUESTION_NAME='Какое из предложенных мест вы бы посетили в первую очередь?'
FIRST_CAPTION='Дикие горы?'
SECOND_CAPTION='Солнечные тропические пляжи?'
THIRD_CAPTION="Или исторические места?"
```

Закончите процесс запуском:

```
marco => test_generator.sh sample_multiple_choice.odt my_holyday_data_1 Holiday_Pictures/*
```

Разве не круто? Теперь, чтобы сгенерировать 1000 подобных файлов, осталось только запустить приведенный ниже скрипт в цикле.

```
Листинг 1: test_generator.sh
WORK_DIR=odt_test_generator_temp_dir

rm -rf $WORK_DIR
mkdir $WORK_DIR
FILENAME=`basename $1 .odt`

cp $1 $WORK_DIR/my_template.odt
cp $2 $WORK_DIR/my_data.sh
shift # удаляем $1 из списка аргументов
shift # удаляем $2 из списка аргументов

## копируем все картинки в рабочий каталог
touch $WORK_DIR/new_pictures_list
for VAR in "$@"
do
CURRENT_FIG=`basename $VAR`
cp $VAR $WORK_DIR/
echo "cp ../$CURRENT_FIG " >> $WORK_DIR/new_pictures_list
done

## подготовка
cd $WORK_DIR
mkdir work
mv my_template.odt work
cd work
source ../my_data.sh
unzip my_template.odt > /dev/null
rm my_template.odt

## заменяем текстовые строки
sed "s|Test Name|$TEST_NAME|" content.xml \
| sed "s|Question|$QUESTION_NAME|" \
| sed "s|Test description and general conditions|$DESCRIPTION|" \
| sed "s|first caption|$FIRST_CAPTION|" \
| sed "s|second caption|$SECOND_CAPTION|" \
| sed "s|third caption|$THIRD_CAPTION|" \
> custom_content.xml
```

»

» Пропустили номер? Узнайте на с. 103, как получить его прямо сейчас.

```
38 mv custom_content.xml content.xml
39
40 ## получаем имена внедренных картинок и заменяем их
41 tr ">" "\012" < content.xml | grep 'draw:image xlink:href' | cut
'-d"' -f2 > ../pictures_list
42 paste ../new_pictures_list ../pictures_list > ../copy_pictures
43 source ../copy_pictures
44
45 ## архивируем все, переименовываем в файл .odt и очи-
щаем
46 find . -type f -print0 | xargs -0 zip ../$FILENAME > /dev/null
47 cd ..
48 mv $FILENAME.zip ../new_$FILENAME.odt
49 cd ..
50 rm -rf $WORK_DIR
```

Скрипт **test_generator.sh** принимает в качестве аргумента шаблон в ODF, ASCII-файл, содержащий текстовые строки, и все изображения, которые следует поместить в новый документ. Первые 10 строк определяют временный каталог и копируют в него все необходимые файлы. Команды **shift** в строках 9 и 10 удаляют шаблон ODF и файл с данными из переменной, хранящей список аргументов, **$@**. Это необходимо для создания цикла в строках с 14 по 19, обрабатывающих только графические файлы. Здесь создается локальная копия каждой картинки, но более важная часть находится в строке 18: она генерирует файл **new_pictures_list**, имеющий вид наподобие

```
marco => cat new_pictures_list
cp ../01_ubuntu_logo.png
cp ../02_mandriva_logo.png
cp ../03_fedora_logo.png
```

Скоро вы узнаете, зачем нужно создавать такой файл, а пока вернемся к коду. Строки с 22 по 28 перемещают нас в каталог **work**, загружают набор переменных из файла данных (строка 26), распаковывают шаблон, а затем, наконец-то, мы можем приступить к генерации нового ODF-файла.

Обратите внимание на раздел замены текстовых строк (строки с 30 по 38), состоящий из нескольких команд **sed**, соединенных конвейером. Каждый вызов **sed** заменяет одну строку текста-местозаполнителя в файле **content.xml** содержимым одной из переменных, входящих в набор, определенный в файле **my_test_data_1**. Если вы позднее захотите приспособить данный скрипт для создания других документов, то это именно тот раздел кода и файл с данными, которые следует изменять.

После создания нового файла **content.xml** исходные картинки заменяются на те, которые мы хотим поместить в шаблон. Это делается за два шага. Строка 41 использует команды **tr** и **grep** для извлечения и записи имен всех картинок в файл **pictures_list**. Имена содержатся в XML-атрибуте xlink внутри **content.xml**, и в результате должно получиться нечто вроде

```
marco => cat pictures_list
Pictures/100000000000008F0000008A1DC84E9A.png
Pictures/10000000000000910000008EDC1B9151.png
Pictures/10000000000000810000008C06E4E423.png
```

После этого остается лишь склеить **new_pictures_list** и **picture_list** вместе, строка за строкой (строки с 42 по 43), чтобы получить командный файл, формируемый в строке 43:

```
marco => cat copy_pictures
cp ../01_ubuntu_logo.png Pictures/100000000000008F0000008A
1DC84E9A.png
cp ../02_mandriva_logo.png Pictures/10000000000000910000000
8EDC1B9151.png
cp ../03_fedora_logo.png Pictures/10000000000000810000008C
06E4E423.png
```

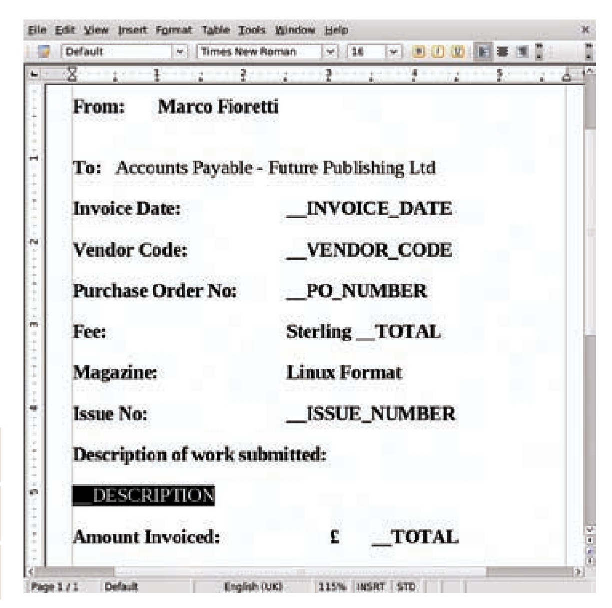
Теперь все части готовы. В остатке скрипта все просто запаковывается, а расширение результирующего файла меняется на ODT. В порядке второго примера давайте рассмотрим, как создавать счета в формате ODF. О том, как это сделать, статей уже полно, но во всех требуют запуска *OOo* и ручной правки. А мы хотим, чтобы компьютер выполнял всю работу за нас. Процедура похожа на уже обсужденную, только проще, потому что нет картинок. Для начала настройте шаблон, а затем создайте такой файл с данными:

```
marco => cat my_invoice_data_file
INVOICE_DATE='2009/03/20'
VENDOR_CODE='007'
PO_NUMBER='LXF 10541'
TOTAL=100
ISSUE=150
DESCRIPTION='Не иначе как за лучший из учебников Linux
Format!'
```

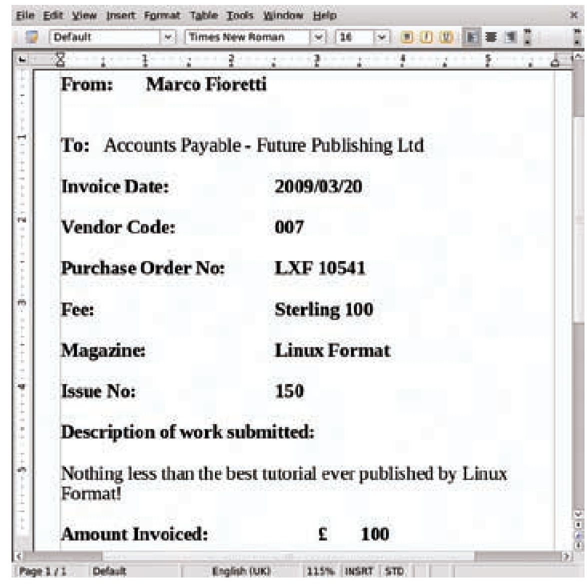
для преобразования шаблона в счет.

Теперь запустите скрипт, приведенный ниже – он получен из сценария оболочки **test_generator.sh**, рассмотренного в первом примере:

```
Листинг 2:
1 WORK_DIR=odt_invoice_generator_temp_dir
2
3 rm -rf $WORK_DIR
4 mkdir $WORK_DIR
5 FILENAME=`basename $1 .odt`
6
7 cp $1 $WORK_DIR/my_template.odt
8 cp $2 $WORK_DIR/my_data.sh
9
10 ## Подготовка
11 cd $WORK_DIR
12 mkdir work
13 mv my_template.odt work
14 cd work
15 source ../my_data.sh
16 unzip my_template.odt > /dev/null
17 rm my_template.odt
18
19 ## заменяем текстовые строки
20 sed "s|__INVOICE_DATE|$INVOICE_DATE|" content.xml \
21 | sed "s|__VENDOR_CODE|$VENDOR_CODE|" \
22 | sed "s|__PO_NUMBER|$PO_NUMBER|" \
```



› **Задачи разные, решение одно: простой шаблон — все, что нужно для генерации счетов компьютером.**


File Edit View Insert Format Table Tools Window Help

Default | Times New Roman | 16

From: **Marco Fioretti**

To: Accounts Payable - Future Publishing Ltd

Invoice Date:	**2009/03/20**
Vendor Code:	**007**
Purchase Order No:	**LXF 10541**
Fee:	**Sterling 100**
Magazine:	**Linux Format**
Issue No:	**150**

Description of work submitted:

Nothing less than the best tutorial ever published by Linux Format!

Amount Invoiced: **£ 100**

Page 1 / 1 | Default | English (UK) | 115% | INSRT | STD

› Результат ваших трудов: счет, готовый к печати или отсылке электронной почтой.

```
23 | sed "s|__TOTAL|$TOTAL|g" \
24 | sed "s|__ISSUE_NUMBER|$ISSUE|" \
25 | sed "s|__DESCRIPTION|$DESCRIPTION|" \
26 > custom_content.xml
27 mv custom_content.xml content.xml
28
29 ## запаковываем все, меняем у полученного файла расши-
рение на .odt и очищаем
30 find . -type f -print0 | xargs -0 zip ../$FILENAME > /dev/null
31 cd ..
32 mv $FILENAME.zip ../new_$FILENAME.odt
33 cd ..
34 rm -rf $WORK_DIR
```

Многое из этого должно быть вам знакомо по описанию Листинга 1, но приглядимся к строкам с 19 по 25. Как уже упоминалось, при изменении шаблона надо добавить команду **sed** для каждой заменяемой строки. Если одна и та же строка встречается несколько раз, как это имеет место для нашей итоговой суммы, не забудьте добавить к **sed** опцию **g** (глобальный), в противном случае будет заменено только первое вхождение данной строки (см. строку 23).

А что с версиями MS Office?

В идеальном мире все перейдут на OpenDocument и безбумажные офисы, и в этом раю не надо будет вникать в форматы файлов, используемых другими людьми, или в лицензии, разрешающие открытие таких документов. Но пока эти деньки не наступили (эх…), не обойтись без распечатки файлов или их конвертирования для бедолаг, прикованных к приложениям, которые понимают только форматы *Microsoft Office*.

К счастью, и конвертировать, и печатать можно автоматически. Однако, в отличие от генерации ODF-файлов, на этом шаге необходим *OpenOffice.org*. Требуется добавить к вашему скрипту строку, которая будет запускать *OpenOffice.org* без графического интерфейса, затем выполнять макрос *OOo* для конвертирования ODF-файла в форматы PDF (для печати) или DOC. Для достижения этой цели имеется несколько макросов; два лучших – SaveAsPDF и SaveAsDoc, с сайта www.xml.com/pub/a/2006/01/11/from-microsoftto-openoffice.html. Альтернативный макрос генерации PDF имеется на сайте http://linux.derkeiler.com/Mailing-Lists/Fedora/2008-06/msg00561.html.

Чтобы распечатать PDF-файл из скрипта, его можно просто «скормить» команде lPr. Использование *OpenOffice.org* в командной строке описано в http://tinyurl.com/rybr9d, но я, так и быть, покажу вам корректный синтаксис:

```
soffice -invisible macro://путь-к-макросу($FILE)
```

Опция **-invisible** заставляет *OpenOffice.org* запускаться без графического интерфейса. Обрабатываемый файл должен передаваться как аргумент (**$FILE**) макроса.

Что мы узнали?

На данном уроке мы изучили метод автоматизации повторяющихся задач с документами, имеющий несколько важных преимуществ. Прежде всего, он работает без запуска *OpenOffice.org* (за исключением печати), то есть его можно выполнять на сервере. Метод также не опирается ни на какие реляционные базы данных; а кроме этого, весьма прост! Во многих случаях, когда XML-инструменты (вроде описанных во врезке *Ресурсы* внизу) слишком сложны для изучения и установки, использование приведенных здесь уловок поможет хотя бы избежать потери времени на повторяющееся редактирование. По сути, кто угодно, имея начальный уровень знаний о сценариях оболочки, может генерировать, изменять и анализировать сотни текстовых ODF-документов, покодировав пару минут.

Но приведенный процесс привлекателен не только благодаря легкости использования. Во-первых, основы данного процесса по идее достаточно хорошо вам знакомы. Мы считаем, что большинство пользователей Linux комфортнее чувствуют себя со скриптами *Bash*, чем в StarBasic – языке написания макросов *OOo*. Что приводит нас к основной причине, по которой мы избегаем макросов *OOo*: реализация логики решения вне *OpenOffice.org* придает намного большую силу, гибкость и потенциал интеграции с другими инструментами, от заданий *Cron* до массовых рассылок и обработки графики. Например, можно запросто добавить в первый скрипт инструменты *ImageMagick* (**LXF116—117**) и масштабировать и кадрировать изображения, прежде чем помещать их в ODF-документ!

Понятно, два представленных скрипта не идеальны. Они не особо гибкие или расширяемые, и в данном виде не совсем надежны. Прежде всего, здесь нет обработки ошибок: если один из файлов отсутствует, или генератору тестов выдано больше картинок, чем имеется в исходном шаблоне, то все пойдет наперекосяк, без всякого сообщения о случившемся.

Вы, конечно, можете улучшить их, но цель данной серии не в разработке промышленных решений. Прежде всего, наш урок доказывает, что OpenDocument не имеет скрытых глубин: ODF-файл – обычный zip-архив, содержащий простой текст и изображения, совместимые с любыми программами. Храните свои данные в формате ODF, и их будет легко восстановить, даже если завтра *OpenOffice.org* исчезнет с лица земли. Во-вторых, что более важно, мы показали вам: ODF можно быстро и легко подстроить под ваши нужды, и с ним можно экспериментировать. LXF

Ресурсы

Если вы намерены серьезно изучить обработку ODF, трюков данного учебника для ваших нужд будет недостаточно. В этом случае обратитесь к XML-инструментам типа Odfpy (http://odfpy.forge.osor.eu) или к книге Дэвида Айзенберга [J David Eisenberg] *OpenDocument Essentials*: ее можно купить по адресу www.lulu.com/content/207835 или прочесть онлайн на http://develop.opendocumentfellowship.com/book.

» **Через месяц** Мы рассмотрим обработку данных в ODF-файлах.

OOo: Вам Math,

Настоящие ученые используют *LaTeX*, но если вы не в восторге от его синтаксиса, то будете рады узнать, что в *OpenOffice.org* можно делать почти то же самое. **Дмитрий Смирнов** расскажет, как именно.



Наш эксперт

Дмитрий Смирнов
Член русскоязычного сообщества *OpenOffice.org*, соавтор переводного издания книги «Руководство пользователя OpenOffice.org 2», а также «OpenOffice.org pro для профессионала» и «OpenOffice.org. Теория и практика».

На этом уроке мы займемся, пожалуй, самым малоосвещенным в широкой прессе компонентом *OpenOffice.org* – программой для набора математических формул *OpenOffice.org Math*. Она не является насущной необходимостью для всех и каждого; ее пользователи – люди, имеющие дело с документами, где применяются математические формулы и выражения: студенты, профессора, технические специалисты, инженеры.

OOo Math (или просто *Math*) может работать как отдельное приложение, но на практике его используют для вставки формул во внешние документы. В этой статье мы рассмотрим ввод формул, а также затронем вопросы их вставки в *OOo Writer*.

Мама, что это?

Именно такой реакции следует ожидать от человека, знакомого с *Microsoft Office* и впервые решившего вставить формулу *Math* в текстовый документ *Writer*. И действительно: вместо привычного ввода символов непосредственно в конкретное место документа создается какая-то рамочка, а внизу появляется еще какое-то поле ввода. Пользователь инстинктивно нажимает на кнопки с шаблонами формул, но вместо привычного появления символов видит какие-то квадратики, да еще и в нижнем поле ввода печатаются какие-то непонятные буковки. И что с этим делать?

А ларчик открывается просто: *Math* использует *командный* ввод формул, то есть пользователь набирает их с клавиатуры, как в языке программирования. Привычное щелканье мышью по панели Выбор всего лишь вставляет шаблон желаемой формулы в поле ввода. Так что пользователю *MSO*, привыкшему к *Equation* и *MathType*, придется учиться заново, так же как и новичку. Пользователям *LaTeX* будет проще.

Когда необходимость обучения «с нуля» становится ясна (а это происходит почти сразу после попытки набрать первую формулу), в голову приходят мысли: «А зачем это надо?» и «Почему не сделали как в *Word*?» Пользователя можно понять: на обучение надо тратить силы и время. Однако в данном случае эти затраты могут с лихвой окупиться в будущем. За счет чего? По сравнению с «обычным» методом, используемым в *Equation*, командный имеет одно важное преимущество: по окончании обучения можно будет вводить формулы, практически не используя мышь, с помощью одной клавиатуры! Представьте себе, что вам больше не понадобится постоянно отвлекаться на панель символов, дабы ввести греческую букву или особенный знак. Так что, может быть, стоит потратить немного времени сейчас, чтобы экономить его потом?

Уговорил? Ну тогда поехали!

С чего начать?

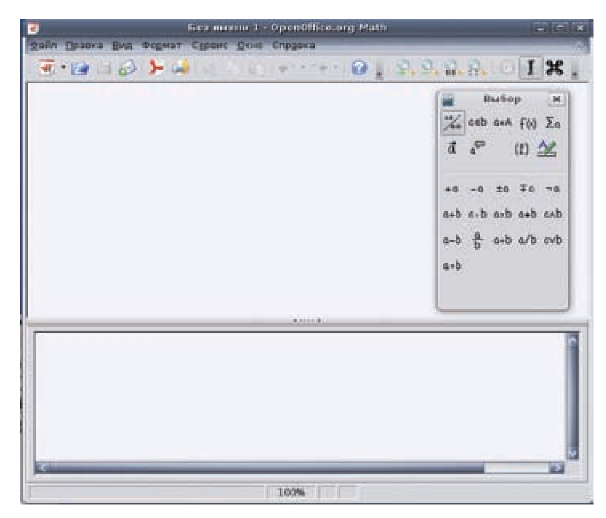
Полюбоваться на изначальный внешний вид *Math* можно, запустив его из меню программ. Точное местонахождение редактора формул различается в зависимости от используемой операционной системы и дистрибутива. В любом случае, появится довольно стандартное окно. Ничего принципиально нового в интерфейсе разработчики придумывать не стали. Все те же меню и панели инструментов. Разница лишь в рабочей области, которая состоит из поля отображения формул (вверху) и поля ввода команд (внизу). В поле ввода команд осуществляется ввод выражений на языке разметки *Math*, а в поле отображения формул появляется математическое выражение, соответствующее введенным командам. При желании, их можно также называть программистским термином «код».

В процессе освоения редактора формул можно запускать *Math* как отдельное приложение, однако набирать в нем текстовые документы не рекомендуется. В большинстве случаев *Math* используется для создания и редактирования математических выражений во внешний программах (преимущественно *OOo Writer*, с которым имеется тесная связь). Впрочем, используя *Math* автономно, вы сможете сохранять набираемые формулы, создавая тем самым их личную библиотеку.

Существуют два подхода ко вводу формул: с помощью шаблонов команд (кода) и вручную. Как правило, первый используется при ознакомлении с *Math*, а также при поиске команд, вид которых неизвестен. После обучения пользователь чаще всего переходит на ручной ввод команд, поскольку это позволяет работать, не отвлекаясь на панели инструментов и меню. Шаблоны команд можно получить, щелкнув на нужную кнопку в панели инструментов Выбор или из контекстного меню поля ввода команд. Чтобы отобразить или скрыть панель Выбор, найдите в главном меню *Math* пункт Вид > Выбор; чтобы отобразить контекстное меню поля ввода команд, щелкните по полю правой кнопкой мыши.

Таким образом, оба подхода образуют три способа ввода формул:

1 с помощью панели инструментов Выбор (или, что то же самое, окна Выбор);



› **Главное окно *OpenOffice.org Math* ничего особенного собой не представляет.**

коллега!

2 с помощью контекстного меню поля ввода команд;
3 вручную.

Рассмотрим использование всех трех способов на простых примерах. Допустим, надо ввести выражение «*a* плюс *b*». Последовательность действий:

1 В верхней половине окна Выбор щелкаем на кнопке Унарные/бинарные операторы, а в нижней половине – на кнопке Сложение +. В поле ввода команд появляется текст <?> + <?>. Заменяем первый набор символов <?> на a, а второй – на b.

2 Из контекстного меню поля ввода команд выбираем Унарные/бинарные операторы – a + b. Как и в предыдущем случае, в поле ввода команд появляется текст <?> + <?>. Аналогично заменяем первый <?> на a, а второй – на b.

3 Печатаем a+b в поле ввода команд.

Заметим, что пробелы, окаймляющие с двух сторон знак +, необязательны.

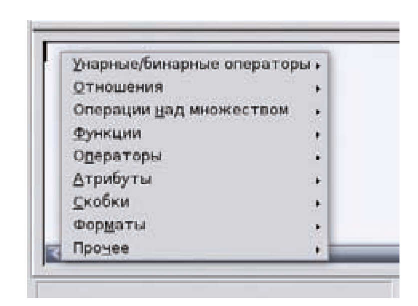
Вы скажете, что про знак «плюс» можно было бы и догадаться. А как поступить в случае, когда ответ не столь интуитивен – например, «*a* умножить на *b*»? Ведь ни точки по центру строки, ни крестика на клавиатуре нет, звездочкой принято обозначать свертку, а буква *x* – неадекватная замена крестику. В таком случае, мы ищем подходящий внешний вид формулы в панели Выбор. После нажатия на соответствующие кнопки, в поле ввода команд появятся шаблоны <?> cdot <?> для точки и <?> times <?> для крестика. Как и в ранее описанном случае, <?> надо заменить на нужные символы.

После ряда подобных действий вы станете все реже обращаться к окну Выбор – ведь это увеличивает время ввода формулы – и будете чаще набирать формулы «от руки» в поле ввода команд.

Вербальная логика

Когда человек сталкивается с чем-то новым, он часто ищет аналогии; случай с *Math* – не исключение. Так на что похож редактор формул *Math*? Пользователь Linux скажет – на командную строку, программист – на язык программирования, лингвист – на измененный английский язык с другим применением, наборщик – на комбинацию цифр и букв. И все они правы. Но важно то, что у метода набора формул в *Math* есть логика.

Когда я изучал программирование, то встречал фразы вроде «сравните … с английским языком». Давайте посмотрим, насколько *Math* может быть сравним с тем же английским? А поскольку английский язык является для нас иностранным, то будем проводить аналогии и с русским языком.



› Контекстное меню поля ввода команд.

Представим себе такую ситуацию: вы хотите набрать формулу с помощью слов. Вы диктуете некоторый текст – а он превращается в формулу. Важное замечание: надо проговаривать этот текст подробно. Например, если учитель математики говорит: «Запишите функцию: эф от икс равно икс-квадрат плюс два икс плюс один», то вы, записывая ее к себе в тетрадь, мыслите примерно так: «Эф, открывающая круглая скобка, икс, закрывающая круглая скобка, равно, икс, степень – два, плюс два икс плюс один». При этом вы пишете нужные символы один за другим, особо не задумываясь. Возможно, вы думаете иначе, например: «Эф, парные скобки, внутри икс...» Это – с лингвистической точки зрения.

Команды *Math* можно вводить похожим образом, как бы переводя их с русского языка на язык разметки *Math*. Если рассмотреть указанный выше пример, то перевод можно осуществить по такой логике:

эф – *f*
открывающая круглая скобка – (
икс – *x*
закрывающая круглая скобка –)
равно – =
степень – ^
плюс – +

Соединив эти команды воедино, получим: f(x)=x^2+2 x+3 (справа – код, слева – результат):

$$f(x)=x^2+2x+3$$

f(x)=x^2+2 x+3

Пробел перед *x* нужен в силу особенностей языка разметки, но во многих случаях вы вольны расставлять пробелы по собственному желанию. Можно записать, например, так: f(x) = x^2 + 2 x + 3, f(x) = x^2+2 x+3 и так далее.

Поговорим теперь о перестановке слов в языке или о перестановке слов в мыслях. Когда вы записываете дробь, вы можете думать: «Числитель, горизонтальная черта, знаменатель» или «Горизонтальная черта, вверху – числитель, внизу – знаменатель». В *Math* можно реализовать оба подхода. Рассмотрим простой пример:

$$\frac{1}{2}$$

Первая логика: «Один, горизонтальная черта, два». Или «Единица над двойкой».

«Перевод» первой логики: один – 1, горизонтальная черта – over, два – 2. Вместе: 1 over 2. Вводятся все символы этой строки команды подряд.

Вторая логика: «Горизонтальная черта, сверху – 1, снизу – 2».

«Перевод»: 1 over 2. Да, в этом случае итоговая строка команды одна и та же. Но прелесть в том, что можно сначала написать over, затем написать слева от over единичку, а справа – двойку, или наоборот. Нет жестких правил порядка ввода команд, важен лишь их итоговый порядок!

Чем это удобно? Если я ввожу длинное выражение в скобках, то мне удобно сначала поставить обе скобки, а затем – вставить нужное выражение между ними. Но я могу сделать и по-другому: ввести все подряд, а не расставлять скобки в самом начале. Кому как удобнее. Появляется бо́льшая свобода ввода. »

› Панель Выбор: отсюда можно вводить шаблоны команд.

» **Пропустили номер?** Узнайте на с. 103, как получить его прямо сейчас.

Итак, я уже указал на то, что *Math* устроен логично. Помимо этого, *Math* гибок и универсален. Некоторые конструкции похожи друг на друга, сами команды у близких операций схожи между собой, а заложенные возможности форматирования позволяют создавать красиво оформленные формулы средствами самого редактора.

Использование скобок

С некоторыми простыми выражениями мы познакомились, теперь усложним задачу. Снова обратимся к лингвистической стороне вопроса. Рассмотрим более сложный пример – «дробь: в числителе 1, в знаменателе 2+3». В этом случае ход мыслей такой: «единица над двойкой плюс три». Но если записать 1 over 2+3, то получится так:

$$\frac{1}{2}+3$$

А это совсем не то, что нам нужно! Для таких случаев используют операторные (групповые) скобки. В Math это фигурные скобки { и }. Тогда ход мыслей может быть, например, такой: «единица над двойкой-плюс-тройкой». Своего рода логическое разделение одной части фразы от другой. Правильная запись: 1 over {2+3}.

Операторные скобки можно использовать не только по их прямому назначению, но и «для себя»: дабы не запутаться в сложном выражении, можно «обернуть» в них его части. При этом главное – не нарушить вид вашей формулы, и количество открывающих операторных скобок должно быть равно количеству закрывающих. Впрочем, это касается всех скобок: круглых – (), квадратных – [], треугольных (угловых) – <> и так далее.

Кстати о вводе скобок. Если круглые и квадратные скобки вводятся с клавиатуры нажатием соответствующих клавиш, то треугольные, фигурные и другие виды реализуются с помощью специальных команд. Например:

$()[]\langle\rangle\{\}$ — `() [] langle rangle lbrace rbrace`

Важный момент: скобки обязательно должны быть парными. Если вам нужна одиночная скобка – вставьте ее как текст, заключив в двойные кавычки. Так, ввод **(** приведет к ошибке, а “(“ отобразит открывающую фигурную скобку в формате, указанном для текста (настроить форматирование формул можно, выбрав нужные пункты из меню Формат).

Помимо простых скобок, в Math есть еще и масштабируемые. Обычные скобки превращаются в масштабируемые путем приписывания ключевых слов left и right к соответствующим командам. Например:

$[\![\frac{a}{b}]\!]$ – немасштабируемые скобки — `ldbracket a over b rdbracket`

$\left[\!\left[\dfrac{a}{b}\right]\!\right]$ – масштабируемые скобки — `left ldbracket a over b right rdbracket`

А как поступить, если надо поставить одиночную масштабируемую скобку? Ведь необходимо же соблюсти парность! Разработчики предусмотрели эту возможность, введя «невидимую» скобку, описываемую ключевым словом none.

$\left\{\dfrac{a}{b}\right.$ – масштабируемые скобки — `left lbrace a over b right none`

Этот «фокус» можно использовать при записи систем уравнений. Подробности можно прочитать в моей статье «Сложная разметка и хитрости в OpenOffice.org Math» (http://i-rs.ru/Stat-i/Konkurs-2007/konkursnye-raboty-53/Slozhnaya-razmetka-i-hitrosti-v-OpenOffice.org-Math) или в книге «OpenOffice.org pro для профессионала», выпущенной издательством «ДМК Пресс» в 2008 году. Там же описаны и другие интересные приемы, обеспечивающие эффективность работы в *OpenOffice.org Math* и демонстрирующие гибкость приложения.

Околосимвольные элементы

Math обладает широким спектром околосимвольных элементов. К таковым можно отнести индексы, стрелки и точки над символами и так далее. Запомнить команды легко. Например, lsub, csub и rsub можно расшифровать как «left subindex», «central subindex» и «right subindex» соответственно. Кстати, и со скобками можно провести аналогию.

Средствами *Math* можно создавать различные индексы, включая вложенные. Пользователю также предоставляется широкий набор надсимвольных элементов: разной длины стрелки, линии, точки, «крышка» и так далее. Приведу несколько примеров: вы легко разберетесь в них самостоятельно.

${}^{a}_{d}\overset{b}{\underset{e}{X}}{}^{c}_{f}$ — `X lsup a csup b rsup c lsub d csub e rsub f`

${}^{a}_{d}\overset{b}{\underset{e}{X}}{}^{c}_{f}$ – то же самое, но запись немного короче — `X lsup a csup b^c lsub d csub e_f`

$\vec{a}\ \vec{abc}\ \overrightarrow{abc}$ – почувствуйте разницу — `vec a` `vec abc` `widevec abc`

$\dot{\vec{a}}$ – векторная производная — `dot vec a`

Более сложные операторы

Рассмотрим несколько операторов, в которых могут использоваться три операнда. Вы можете увидеть соответствующие им конструкции, нажав на кнопку Операторы в верхней части окна Выбор. С помощью этих операторов можно создавать формулы, содержащие пределы, интегралы, суммы и произведения.

Общий синтаксис всех этих операторов выглядит так:

`название_оператора from нижний_предел to верхний_предел выражение`

Нижний и верхний пределы являются необязательными и могут существовать отдельно друг от друга. Проиллюстрирую это на примере.

$\int f(x)dx$ – неопределенный интеграл — `int f(x)dx`

$\int\limits_{a} f(x)dx$ – интеграл с нижним пределом. Используется, например, при записи интегрирования по площади — `int from a f(x)dx`

$\int\limits^{b} f(x)dx$ – интеграл с верхним пределом. Не имеет математического смысла — `int to b f(x)dx`

$\int\limits_{a}^{b} f(x)dx$ – определенный интеграл — `int from a to b f(x)dx`

В случае, когда на месте верхнего и нижнего пределов используются выражения, не забывайте про операторные скобки, иначе может получиться не то, что вы хотите. Сравните следующие записи:

$\int\limits_{2a+5}^{4} b+7x^2dx$ — `int from {2 a+5} to 4 b+7 x^2 dx`

$$\int\limits_{2a+5}^{4b+7} x^2\,dx$$ – определенный интеграл

```
int from {2 a+5} to {4 b+7} x^2 dx
```

$$\int\limits_{2} a^2 x^2\,dx$$

```
int from 2 a^2 x^2 dx
```

$$\int\limits_{2a^2} x^2\,dx$$

```
int from {2 a^2} x^2 dx
```

Вместо ключевых слов from и to можно использовать уже знакомые надсимвольные индексы csub и csup соответственно.

Матрицы

На матрицах мы остановимся отдельно. Во-первых, у них довольно сложный синтаксис. А во-вторых, конструкцию, которая создается кодом матрицы, можно использовать не по прямому назначению. Это позволяет создавать эффектные формульные конструкции и лишний раз показывает гибкость *Math*. Но давайте по порядку.

Матрица представляет собой прямоугольный математический объект, в котором данные размещены в строках и столбцах. Некоторые могут назвать матрицу таблицей, а некоторые таблицы (например, в экономике) стали называть матрицами. Действительно, у матриц и таблиц есть много схожего, но есть и отличия. Важное свойство матрицы состоит в том, что количество элементов в строках (столбцах) матрицы одинаково в любой строке (столбце). Для таблиц это выполняется не всегда, поскольку в них могут находиться объединенные ячейки.

Человек читает матрицу слева направо. Это отражено и в математической форме, когда при ссылке на элемент матрицы сначала говорится номер строки, а затем – столбца. Кроме того, в большинстве стран мира (в том числе и в России) чтение идет сверху вниз. Ввод матрицы в *Math* реализован соответствующим образом.

$$\begin{matrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{matrix}$$

```
matrix{a_11 # a_12 # a_13 ## a_21 # a_22 # a_23}
```

Сопоставив код результату, легко понять, что элементы одной строки разделяются между собой решеткой, а сами строки – двумя решетками.

Обычно матрицу записывают в круглых или квадратных скобках. Для корректной записи необходимо использовать масштабируемые скобки.

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$$

```
left [ matrix{a_11 # a_12 # a_13 ## a_21 # a_22 # a_23} right ]
```

Допустим, надо ввести вектор – матрицу-столбец. Есть два способа сделать это. Первый – использовать уже знакомую команду matrix, а второй – взять специальную команду stack. Просто сравните:

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix}$$

```
left [ matrix{a_11 ## a_12 ## a_13} right ]
```

$$\begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \end{bmatrix}$$

```
left [ stack{a_11 # a_12 # a_13} right ]
```

Как я говорил раньше, матрицы можно использовать не по прямому назначению. Одним из красивых примеров, наглядно иллюстрирующим такую возможность, является запись вида «сложение уравнений в системе».

$$\begin{array}{r} +\left\{\begin{array}{rrl} x & +10\,y & =31 \\ -x & +4\,y & =11 \end{array}\right. \\ \hline 14\,y \quad =42 \end{array}$$

```
"+"
left lbrace alignr matrix{
x#+10 y#alignl {}=31##
-x#+4 y#alignl {}=11}
right none newline
alignr overstrike phantom {matrix{
x#+101 y#alignl {}=15qq}} newline
alignl phantom{"+" left lbrace right none}
{alignr matrix {phantom -x#~14 y#alignl
{``}=42}}
```

Похожий на этот и другие примеры описаны в уже упомянутой статье «Сложная разметка и хитрости в OpenOffice.org Math».

Вставка в текст

На первый взгляд, в этом нет ничего необычного. Однако есть некоторые тонкости, которые бывает полезно учитывать при работе.

Для вставки формулы выберите Вставка > Объект > Формула в главном меню *Writer*. Вставленная формула является объектом *Math*, поэтому она имеет те же свойства, что и другие объекты: обтекание, расположение, привязка, обрамление. При этом стиль Формула, который соответствует формулам *Math*, является стилем врезок.

Один из часто задаваемых вопросов звучит примерно так: «Как убрать отступы слева и справа от формулы?» Эта проблема решается просто: из контекстного меню для формулы выбираем Объект, идем на вкладку Обтекание и выставляем нулевые отступы. Более того, это можно сделать сразу для всех формул в документе. Для этого надо открыть окно Стили и форматирование (Формат > Стили или F11), перейти на стили врезок, щелкнуть правой кнопкой на стиле Формула, выбрать из контекстного меню Изменить и выставить нулевые отступы на вкладке Обтекание.

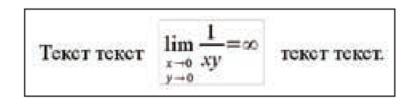
Еще одним важный моментом является размещение формулы в тексте. По умолчанию объект выравнивается по высоте относительно центра. Это может привести к тому, что вставленная формула будет некрасиво смотреться в тексте.

Исправить это можно простым смещением формулы по вертикали, либо с помощью мыши, либо установкой нужных параметров в секции Положение на вкладке Тип диалогового окна Объект (вызывается из контекстного меню формулы, либо Формат > Врезка).

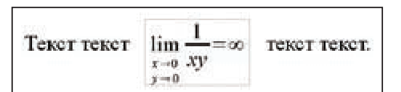
Вопросы совместимости

Часто можно слышать вопрос: «Совместимы ли *Microsoft Equation* и *OpenOffice.org Math*?» Если отвечать кратко, то нет. Это разные редакторы, результатом работы в которых являются разные объекты. Поэтому при открытии документов *Word* в *Writer*, как и при открытии в *Word* документов, созданных в *Writer* и конвертированных в формат *Word*, при отображении формул часто возникают ошибки. По сути, нельзя без проблем «перескочить» с одного редактора на другой, потому что многие формулы придется вводить заново.

Многие издательства принимают документы, содержащие формулы, только в формате *Microsoft Equation*. А что делать тем, кто работал в *OpenOffice.org*? К сожалению, им придется набирать формулы заново. Либо попытаться договориться с издательством об использовании *Math*. LXF



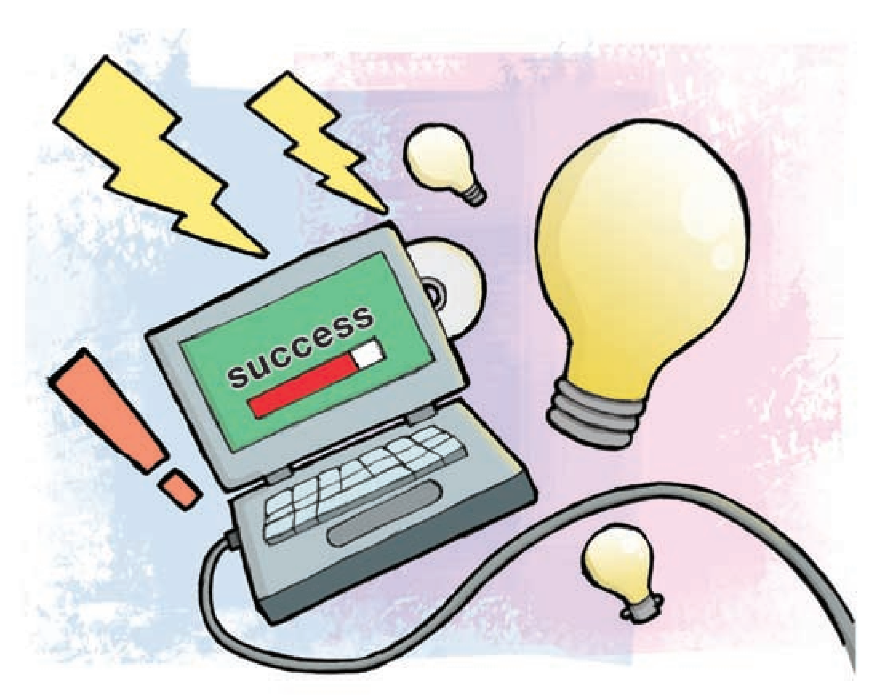
› «Некрасивая» формула: lim как бы вылез из текста.

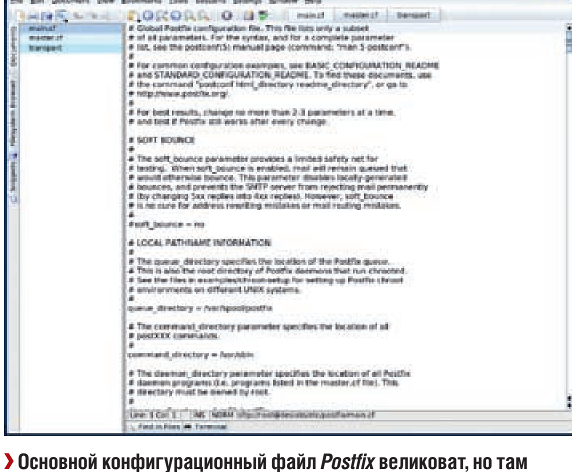


› «Красивая» формула: все слова находятся на одной линии.

Сервер почты:

Часть 8: Нейл Ботвик завершает установку почтового сервера с Postfix, а заодно учит избавляться от спама и вирусов..





› **Основной конфигурационный файл *Postfix* великоват, но там много комментариев, и значения по умолчанию разумные.**



Наш эксперт

Нейл Ботвик
У Нейла Ботвика по компьютеру в каждой комнате. Но по соображениям безопасности он никогда не скажет вам, где центральный сервер.

Еще в **LXF115** мы настроили почтовый сервер IMAP, и можем читать почту откуда угодно. В прошлом месяце мы добавили web-интерфейс, но почта по-прежнему сперва поступает к провайдеру или на другой почтовый сервис, а потом вытягивается оттуда с помощью *Fetchmail*. Пора замкнуть круг, добавив SMTP-сервер, который позволит доставлять почту прямо на IMAP-сервер. Это означает, что ее можно будет отфильтровать на вирусы и спам перед доставкой. По существу, SMTP устанавливает правила, определяющие, как почта будет передаваться между компьютерами.

Основной компонент в этом процессе – MTA (Mail Transport Agent – Агент передачи почты); он отвечает за получение почтовых сообщений с одного компьютера и передачу их на другой. Также можно встретить сокращения MUA и MDA. MUA (Mail User Agent, Пользовательский почтовый агент) – это программа, используемая для управления сообщениями, а MDA (Mail Delivery Agent, Агент доставки почты) отвечает за получение почты с сервера и доставку писем в ящики пользователей. Большинство MTA могут выступать как MDA, но в общем случае применяется отдельная, более продвинутая программа. При сборе почты с *Fetchmail* в качестве MDA мы использовали *Procmail*, с ней и продолжим работать.

У отдельного MDA есть еще одно преимущество: программы можно объединять в цепочки. Вместо прямой передачи почты от MTA к MDA ее можно отправить в спам-фильтр, потом в антивирус, и, наконец, в *Procmail* для доставки пользователю. Тогда все письма перед доставкой будут полностью проверены на нежелательное содержимое и соответствующим образом помечены или отфильтрованы как-то еще.

Для Linux существует несколько популярных почтовых серверов, в том числе *Postfix*, *Sendmail* и *Exim*. И *Sendmail*, и *Exim* – достаточно продвинутые программы; правда, синтаксис конфигурационного файла *Sendmail* так сложен, что мог бы посоперничать с проектом SETI. Но мы в итоге остановились на *Postfix* версии 2.5.6: эта версия вышла в начале текущего года, и ее легко достать. Если она еще не установлена, добавьте ее обычным образом с помощью менеджера пакетов.

Настройка Postfix

В *Postfix* есть несколько больших, хорошо откомментированных конфигурационных файлов в **/etc/postfix**, но в установке по умолчанию нужно изменить совсем немного. Основной файл конфигурации – **/etc/postfix/main.cf**, и прежде чем что-то менять в нем, прочтите комментарии. Теперь, предполагая, что ваш домен – **mydomain.ru** и нужно обрабатывать почту также с **mydomain.com** и **example.com**, изменим файл конфигурации. Начнем со строки

```
proxy_interfaces = 192.168.1.1 # optional
```

Менять **proxy_interfaces** не обязательно, но это защищает от почтовых «петель» (mail loops). Пропустите эту строку, если у вас прямое соединение без маршрутизатора или NAT; в противном случае подставьте IP-адрес своего интернет-шлюза. Теперь установите параметр

» **Месяц назад** Мы добавили к нашему *Apache* web-приложения.

Долой спам!




```
myhostname = mydomain.ru
```

Это имя вашего хоста и имя, которое будет использовать *Postfix* при подключении к другим серверам. Оно должно разрешаться на внешний IP-адрес, иначе некоторые серверы могут отказать в соединении как потенциальному спамеру. Затем измените **mydestination** следующим образом:

```
mydestination = mydomain.ru,mydomain.com,example.com
```

Здесь мы указали разделенный запятыми список доменов, почту с которых будет обрабатывать сервер – получив сообщение с домена, отсутствующего в этом списке, он попробует отправить его серверу, который сможет его обработать. Теперь запишем в **myorigin** адрес вашего домена:

```
myorigin = mydomain.ru
```

Обратите внимание, что теперь вся почта с этого сервера будет выглядеть как почта с этого домена. Домен также добавится ко всем адресам без доменной части (локальная почта).

Сообщения от несуществующих пользователей обычно отклоняются. Параметр **luser_relay** определяет получателя по умолчанию для этих писем. Если вы хотите получать сообщения и для несуществующих пользователей, установите его в значение

```
luser_relay = имя_пользователя # optional
```

Тогда можно использовать разные имена пользователей для разных целей, но возрастет объем спама, так как пользователям со случайно сгенерированными именами спам отправляется в колоссальном количестве.

При отправке почты *Postfix* обычно пытается соединиться с почтовым сервером получателя напрямую. Если ваш провайдер не разрешает такие подключения или диапазон IP-адресов провайдера находится в «черном списке» как спамерский, можно сделать так, чтобы почта проходила сквозь почтовый сервер вашего провайдера. Для этого установите параметр:

```
relayhost = smtp.myisp.ru # optional
```

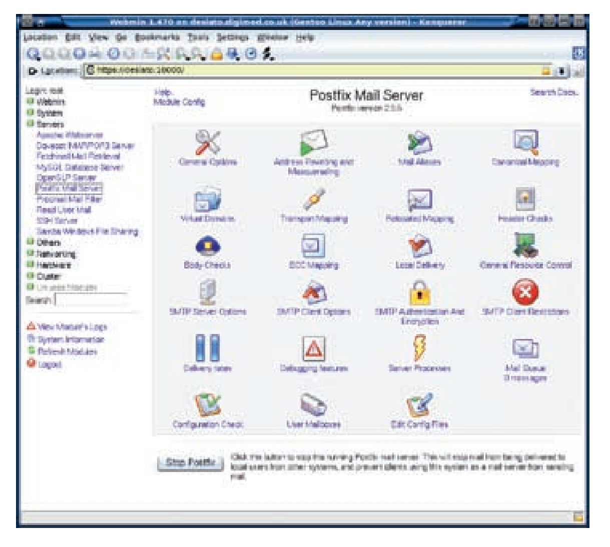
Не исключено, что этот сервер уже указан как SMTP-сервер в настройках вашего почтового клиента. Наконец, установите параметр:

```
mailbox_command = /usr/bin/procmail
```

Postfix может доставлять почту в ящики пользователей напрямую, используя параметр **home_mailbox** или **mail_spool_directory**, но может и работать с внешним агентом доставки почты, что существенно добавляет гибкости. Так как в **LXF115** мы настроили *Procmail* для работы с *Fetchmail*, остановимся на этом варианте (по крайней мере, пока). Если этого номера нет у вас под рукой,

Настройка роутера

Чтобы получать почту извне, вам потребуется статический IP-адрес, который предоставляют многие провайдеры (хотя иногда за дополнительную плату). Понадобится также DNS-запись MX для вашего домена: она скажет другим почтовым серверам, куда доставлять почту для вас. Ее можно получить там же, где вы регистрировали свое доменное имя. SMTP работает через порт 25, и его нужно открыть на маршрутизаторе (он же роутер) и перенаправить на сервер.



› **Это не опции настройки *Postfix*, а группы этих опций: есть из чего выбрать при необходимости.**

добавьте для *Procmail* в **/etc/procmailrc** следующие строки:

```
MAILDIR=/var/spool/mail
DEFAULT=$MAILDIR/$LOGNAME/
LOGFILE=/var/log/procmail
```

После этого запустите (или перезапустите) сервис *Postfix*, настройте свой почтовый клиент на сервер localhost (или имя хоста компьютера, на котором работает *Postfix*, если это другая машина вашей сети) и порт 25 и отправьте самому себе тестовое сообщение. Перед отправкой выполните в терминале команду

```
sudo tail -f /var/log/messages | grep postfix
```

Она профильтрует все новые данные, записанные в файл системного журнала, и выведет на экран то, что относится к *Postfix*. При отправке письма вы должны увидеть несколько строк вывода с сообщением, что соединение с почтовым клиентом было установлено, сообщение получено и доставлено *Procmail*. Если вы видите сообщение об ошибке, оно должно ссылаться на настройку, подлежащую исправлению. Если ошибки нет, а почта не доставлена, вероятно, она успешно попала к *Procmail*, но не сработала доставка. Загляните в файл журнала **/var/log/procmail**.

Авторизованный доступ

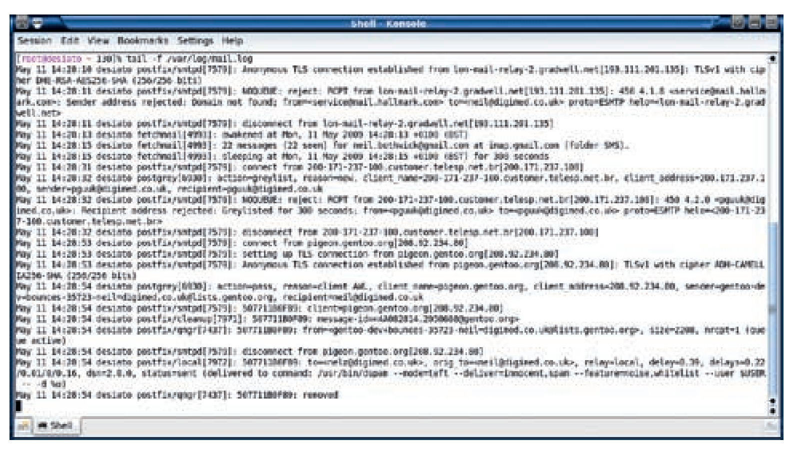
В былые времена, когда обитателями киберпространства были одни ученые да технари, почтовые серверы сотрудничали очень дружелюбно. Любой сервер переслал бы полученное сообщение должному адресату или ближайшему к нему серверу. Все закончилось, когда Сэнфорд «Спамфорд» Уоллес [Sanford 'Spamford' Wallace] и его коллеги сообразили, что это отличный метод массовой рассылки незваных сообщений. Теперь пересылку в почтовых серверах отключают. Это означает, что сервер принимает и доставляет только почту, удовлетворяющую одному из трех критериев:

» Сообщение предназначено для одного из адресатов в списке доменов сервера.

» Сообщение отправлено с компьютера в списке доменов сервера. »

Скорая помощь

Все эти программы – демоны, и обычно ничего не выводят на терминал, на котором запущены. Весь полезный вывод отправляется в файл журнала: либо их собственный, либо через *syslog*. Для просмотра содержимого журнала в реальном времени при настройке используйте команду **tail -f**.



› **Следите за журналами при проверке новых возможностей — ошибки обычно бросаются в глаза.**

» Пользователь, отправляющий сообщение, аутентифицировался перед отправкой почты.

Первые два пункта настроены по умолчанию в любом нормальном MTA, то есть почта принимается, только если получатель или источник находится в сети сервера. Поэтому может понадобиться изменить адрес SMTP-сервера в почтовом клиенте при переключении из одной сети в другую, так как вы вынуждены использовать сервер этой сети.

Однако аутентификация помогает избежать этой проблемы, потому что она означает, что пользователь «правильный», и сервер перенаправит сообщение ему. На собственном сервере это нужно, если вы хотите иметь возможность отправлять почту откуда угодно: из локальной сети, через беспроводное подключение в кафе или через 3G USB-модем. Для аутентификации *Postfix* использует *Cyrus SASL*, поэтому убедитесь, что он установлен, а сервис *saslauthd* запускается при загрузке. Теперь добавьте такие строки в конец файла **/etc/postfix/main.cf**:

```
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain = $myhostname
broken_sasl_auth_clients = yes # optional
smtpd_recipient_restrictions = permit_sasl_
authenticated,permit_mynetworks,check_relay_domains
```

Строка, отмеченная как опционная, нужна только для некоторых версий *Outlook Express* и *Microsoft Exchange*. Перезагрузив конфигурацию, по локальной сети вы все еще сможете отправлять почту с почтового клиента без аутентификации. У вас также должна быть возможность получать и пересылать почту при подключении извне, если почтовый клиент настроен на аутентификацию при отправке (с теми же именем пользователя и паролем, которые используются для получения почты; это справедливо по умолчанию для большинства клиентов). Подробное руководство имеется по адресу http://postfix.state-of-mind.de/patrick.koetter/smtpauth; из него также можно узнать, как зашифровать трафик с помощью TLS. Это важно при подключении извне: в противном случае ваши пароли можно будет перехватить.

Скорая помощь

Попробуйте вносить небольшие изменения в конфигурационные файлы и проверять их после каждой правки. Так будет гораздо проще отследить ошибку. Здравая страховочная политика – создавать резервные копии файлов перед каждым изменением, для этого подойдет *Subversion* или *Git*.

Фильтрация спама

Тихо ли вы презираете спам или люто его ненавидите, но считаться с ним приходится всем. Обычно есть два варианта: либо фильтруйте почту с помощью спам-фильтров провайдера или почтового сервиса, с риском потерять нормальные письма из-за ложных срабатываний (письма, ошибочно признанные спамом), либо пользуйтесь локальным спам-фильтром. Последняя опция предоставляет больший контроль, но при каждом запуске клиента все равно придется загружать и сканировать всю почту, порождая задержки. Теперь у вас есть альтернатива: пусть почтовый сервер сканирует приходящие сообщения, работая в фоновом режиме! Тогда при запуске клиента нужно будет только загрузить проверенную почту. Выбор здесь богатый: *Spam Assassin* (http://spamassassin.apache.org), *Bogofilter* (http://bogofilter.sourceforge.net)… ну, а мы возьмем *Dspam* с сайта http://dspam.nuclearelephant.com. Все эти программы есть в большинстве репозиториев, и все они используют метод поиска спама под названием «байесовский анализ» (Bayesian analysis), рассматривающий частоту различных слов в сообщениях.

Есть несколько способов вызова *Dspam* из *Postfix* – как фильтр содержимого, как транспорт или как команда почтового ящика. Воспользуемся последним вариантом: он, может, и не самый эффективный, зато самый простой в настройке. Когда фильтрация заработает как нужно, всегда можно заняться и другими методами.

Postfix может вызывать *Dspam* напрямую для каждого письма, но это неэффективно; лучше воспользоваться демоном *Dspam*, запускаемым как сервис при загрузке системы. *Postfix* вызывает *dspamc*, который использует демона вместо запуска отдельного экземпляра *Dspam*. Первый шаг – отредактировать файл **/etc/dspam.conf** (он может находиться в подкаталоге **/etc**) и убедиться, что данные параметры установлены:

```
TrustedDeliveryAgent "/usr/bin/procmail"
UntrustedDeliveryAgent "/usr/bin/procmail -d %u"
Preference "signatureLocation=headers"
ServerMode dspam
ServerPass.Relay1 "secret"
ClientIdent "secret@Relay1"
```

Первые две строки велят *Dspam* пользоваться *Procmail* для доставки почты всем пользователям. Опция **signatureLocation** предписывает *Dspam* хранить сигнатуру сообщения в заголовках, а не в теле письма. *Dspam* добавляет сигнатуру в каждое обрабатываемое сообщение, и поэтому не будет просматривать одно и то же письмо дважды. Последние три опции разрешают клиенту *dspamc* взаимодействовать с демоном. Убедитесь, что демон *Dspam* настроен на запуск при загрузке системы с помощью программы настройки сервисов своего дистрибутива, и перезапустите его после изменения файла конфигурации.

Затем откройте файл **/etc/postfix/main.cf** и установите параметры

```
mailbox_command = /usr/bin/dspamc --client --mode=teft
--deliver=innocent,spam --feature=noise,whitelist --user
$USER -- -d %u
```

которые скажут *Postfix*, соответственно, использовать *dspamc* для доставки почты, обучаться в зависимости от вашего выбора (**mode=teft**), доставлять как нормальные сообщения, так и спам, и от имени какого пользователя доставлять почту. *Dspam* может обрабатывать спам двумя способами: изолировать его в карантине (который можно просмотреть через web-браузер и найти письма, признанные спамом ошибочно, а остальное удалить), или доставлять всю почту, но помечать спам как спам. Это дело вкуса, но я бы предпочел последний вариант: при этом сообщения все еще доступны с любого почтового клиента и уже помещены в отдельную почтовую папку. Отфильтровать сообщения можно по заголовку

```
X-DSPAM-Result:
```

который содержит **Innocent** или **Spam**.

Изменив конфигурационный файл, перезапустите *Postfix* или заставьте его перечитать настройки командой

```
postfix reload
```

> «Байесовские спам-фильтры обучаются на вашей почте.»

Байесовские спам-фильтры узнают, что спам, а что нет, прямо из вашей почты. Они особо удобны благодаря отсутствию заданного набора правил и приспосабливаются к вашим потребностям. Недостаток, правда, в том, что их нужно обучать. Обучение происходит в двух направлениях. Во-первых, нужно дать им на пробу множество сообщений, и спама, и полезных (их иногда называют «хамом» – «ham»), чтобы они поняли, какие вам нужны, а какие не нужны. Поэтому до установки *Dspam* не удаляйте спам, просто поместите его в отдельную папку. Потом можно провести первичное обучение с помощью команды **dspam_train**, например, таким образом:

```
dspam_train username spam_dir ham_dir
```

Dspam фильтрует спам отдельно для каждого пользователя, и у каждого своя база данных классификации. Остальные два аргумента – пути к каталогам со спам- и хам-сообщениями, каждое в отдельном файле. Так как здесь используется стандартный формат почтового каталога IMAP-серверов, можно просто указать пути к папкам «входящие» и «спам» своего сервера, например:

```
dspam_train arthur /var/spool/mail/arthur/.INBOX.spam/cur
/var/spool/mail/arthur/cur
```

Нужно также провести обучение «в деле», особенно в первое время. Оно включает информирование *Dspam* при каждом неверном определении спам-сообщения. Добросовестное отношение к этому вначале – залог успеха в будущем, потому что *Dspam* предполагает, что все неисправленные классификации верны, и принимает это к сведению. Простейший способ информировать *Dspam* о ложных тревогах и просчетах – напрямую из почтового клиента, задав несколько почтовых псевдонимов (алиасов). Они определяются в файле **/etc/mail/aliases**, который уже содержит несколько стандартных псевдонимов, поэтому откройте его на редактирование от имени root и добавьте туда две строки:

```
dspam-spam: "|/usr/bin/dspam --user nobody --source=error
--class=spam"
dspam-notspam: "|/usr/bin/dspam --user nobody --source=error
--class=innocent"
```

Определение псевдонима состоит из имени, двоеточия и поля назначения. Это может быть другой пользователь, полный почтовый адрес или любая их последовательность, разделенная запятыми. Это может быть и команда, предваряемая символом |. Она запустит программу и передаст ей содержимое сообщения в стандартный поток ввода. Теперь можно сообщить о ложных срабатываниях и пропущенном спаме, просто перенаправив сообщения на dspam-notspam@your.domain или dspam-spam@your.domain соответственно. Если ваш почтовый клиент поддерживает макросы, можно связать эти действия с комбинацией клавиш или пунктом меню, значительно упростив процесс обучения *Dspam*. Мы сделали этот псевдоним общим, применив пользователя 'nobody', но можно было бы задать отдельную пару псевдонимов и для каждого из пользователей. После изменения файла **aliases** потребуется выполнить команду

```
newaliases
```

чтобы добавить изменения в базу данных псевдонимов *Postfix*.

Ранняя фильтрация спама

Dspam хорошо обнаруживает спам, но часть спама проще обработать из *Postfix*. Спамеров не волнуют тонкости почтового обмена, поэтому неудивительно, что они почти не вникают в обычные правила передачи почты. Стараясь отправить побольше сообщений, они идут кратчайшим путем, опуская или подделывая информацию, отличающую настоящее письмо. Поэтому огромное количество спама можно отпихнуть на начальной стадии SMTP-обмена, отклонив соединение. При этом экономится трафик, так как почта не загружается, и *Dspam* не должен ее обрабатывать. Некоторые из этих типов некорректных сообщений можно отклонить, добавив строки в файл **/etc/postfix/main.cf**:

```
# HELO restrictions
smtpd_delay_reject = yes
smtpd_helo_required = yes
smtpd_helo_restrictions =
permit_mynetworks,
reject_non_fqdn_hostname,
reject_invalid_hostname,
permit
# Sender restrictions
smtpd_sender_restrictions =
permit_sasl_authenticated,
permit_mynetworks,
reject_non_fqdn_sender,
reject_unknown_sender_domain,
permit
# Recipient restrictions
smtpd_recipient_restrictions =
reject_unauth_pipelining,
reject_non_fqdn_recipient,
reject_unknown_recipient_domain,
permit_mynetworks,
permit_sasl_authenticated,
reject_unauth_destination,
permit
```

Здесь нет места на углубление в детали работы этого кода, но если вы хотите узнать побольше, зайдите на сайт www.postfix.org.

Сканирование на вирусы

Если с сервера будут загружать почту пользователи Windows, гуманно было бы проверить ее на вирусы. *Dspam* может это сделать с помощью *ClamAV*. Во-первых, убедитесь, что *ClamAV* установлен, а сервис *clamd* запущен, потом откройте файл **dspam.conf** и раскомментируйте следующие настройки *ClamAV*:

```
ClamAVPort 3310
ClamAVHost 127.0.0.1
ClamAVResponse spam
```

Первые две строки, скорее всего, менять не надо, а третья говорит *Dspam*, что делать с почтой, если *ClamAV* найдет вирус – **reject** отбросит сообщение и возвратит ошибку, **accept** примет сообщение и потом молча от него избавится, а **spam** велит *Dspam* считать сообщение спамом и изолировать или пометить его.

Теперь у вас должна быть полная почтовая система для SMTP, IMAP и web-почты. Программы, рассмотренные здесь, очень гибкие, и мы охватили только их базовые возможности, поэтому засучите рукава, прочтите документацию и настраивайте ПО в соответствии с вашими нуждами. LXF

Журналы Postfix

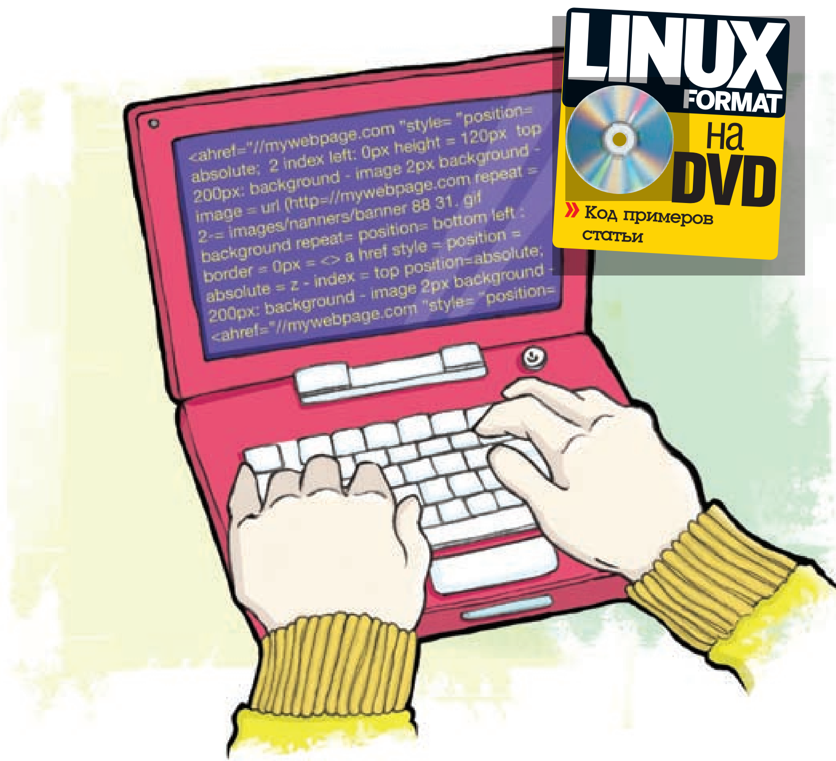
Postfix записывает все в *syslog* и может завалить вас высылаемой информацией. Обычно лучше всего приказать вашей службе журналирования отправлять эти сообщения в отдельный файл. Если вы пользуетесь **syslog-ng** (как в большинстве дистрибутивов), добавьте в файл **/etc/syslogng/syslog-ng.conf** такие строки:

```
#Postfix logging
destination mail { file("/var/log/mail.log"); };
filter f_mail { facility(mail); };
filter f_notmail { not facility(mail); };
log { source(src); filter(f_mail); destination(mail); };
```

» **Через месяц** Расширим охват сетью через Virtual Private Networking.

Python: Создаем

Часть 2: Думаете, жизнь может быть лучше? Она и станет лучше, если вы, следуя **Нику Вейчу**, создадите себе бота – чтоб вкалывал за вас.





Наш эксперт

Ник Вейч
Ник Вейч запустил *Linux Format* и редактировал его первые восемь лет. А потом устал и ушел.

Представьте, как упростилась бы наша жизнь при наличии оравы слуг, готовых выполнять наши приказы: они приносили бы почту, чистили клетку с гиппогрифом, пинали Майка… ну, всякое такое. Увы, мы живем в слишком раннем историческом периоде, когда еще не внедрены дешевые, эффективные и послушные роботы-слуги, и приходится все делать самим. Другой способ отлынивать от работы – завести виртуальную прислугу. О да, под это определение может попасть практически любое ПО, но я-то имею в виду простой и удобный интерфейс, способный выполнять полезные задачи и сообщать вам то, что вы хотите знать. Если он сумеет ставить в тупик ваших друзей и убеждать ваших врагов пересмотреть свою матрицу враждебности, тем лучше.

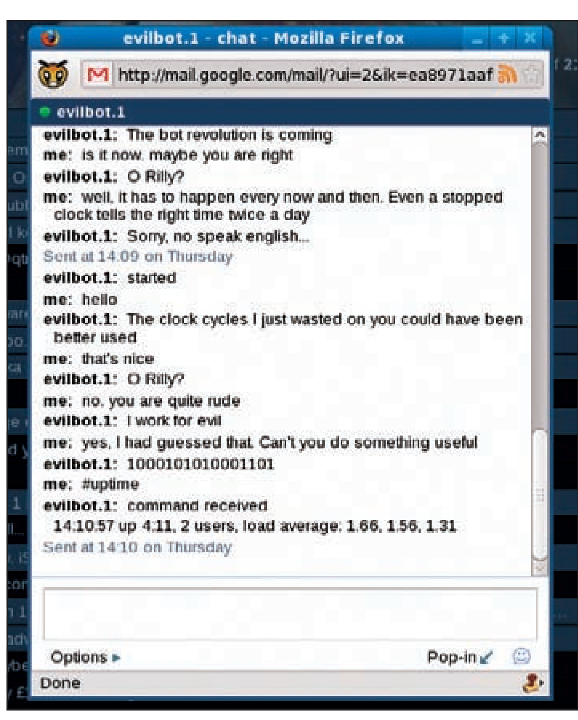
Нечасто используемым, но удобным способом общения с подобными существами является чат. Стоит ли связываться с хитроумными SSH-туннелями или нудными сервисами на базе web, если можно легко общаться с помощью средства, из которого вы и так не вылезаете? В свете вышесказанного, наш маленький подручный будет чат-ботом: пусть сидит на канале чата и ждет, когда хозяин подаст голос; или, допустим, при случае проявляется и сообщает вам о чем-нибудь.

Я Xmpppy

Протокол Jabber/XMPP отлично поддерживается Python. Он реализован в рамках всеобъемлющего сетевого модуля *Twisted*, но есть и более легковесная версия, подходящая для наших нужд, по имени *Xmpppy*. В вашем дистрибутиве наверняка имеется этот пакет, но можно скачать код Python с сайта http://xmpppy.sourceforge.net.

Чтобы разобраться в *Xmpppy*, начнем с примеров командной строки. Но сперва обзаведемся тестовой учетной записью: разве ваш бот не заслужил собственного имени пользователя в Google? Для тестирования вам понадобятся как минимум два идентификатора Jabber ID. Итак, мы зарегистрировали учетную запись Gmail специально для бота, вошли через web-браузер и пригласили другого пользователя Gmail в чат. Можно было бы настроить все это в самом *Xmpppy*, но так будет сложнее: для первичного тестирования лучше иметь две учетных записи, которые могут общаться друг с другом. После этого наберите в оболочке **python** для доступа к интерактивной командной строке Python:

```
>>> import xmpp
>>> jid=xmpp.protocol.JID("botaddress@gmail.com")
```



› Под нашим чутким руководством, ваш чат-бот скоро будет готов к злодеяниям.

» **Месяц назад** Мы заставили рабочий стол предсказывать погоду.

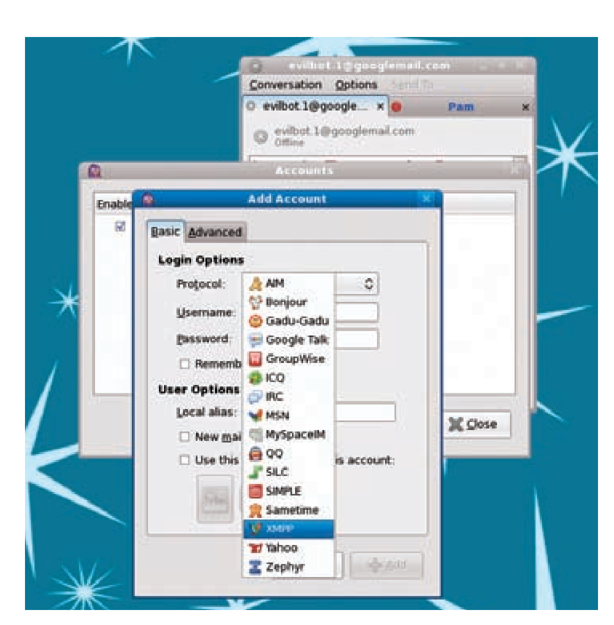
говорящего бота

Это мы установили ID Jabber. Он сильно смахивает на адрес электронной почты, потому что мы для целей нашего урока взяли учетную запись Google; однако любой другой Jabber ID тоже подойдет. Он состоит из двух частей (а может и из трех, но об этом потом): имя пользователя и домен. Домен – это место, где *Xmpppy* будет искать сервер.

Создав пользователя, надо создать и экземпляр клиента. Клиент в *Xmpppy* – это объект, который контролирует соединение, обрабатывает сообщения и вообще взаимодействует с сервером. Клиент и его подключение создаются за несколько шагов: сначала нужно создать экземпляр клиента (в качестве аргумента потребуется Jabber ID), а затем мы попробуем соединиться с сервером. Установив соединение, нужно аутентифицироваться для выполнения дальнейших действий.

```
>>> myclient=xmpp.Client(jid.getDomain(),debug=[])
>>> myclient.connect()
'tls'
>>> myclient.auth(jid.getNode(),'botpasswd')
'sasl'
```

Здесь нужно обратить внимание на пару моментов. Сперва мы создали экземпляр клиента и вызвали метод **jid.getDomain** для получения имени сервера из созданного нами объекта **jid**. Далее, вас просят еще указать уровень отладки, определяющий качество обратной связи, которую вы получаете. Мы установили свой на пустой список, но если вам охота завалить себе экран сообщениями, добавьте в него строку **always**. Впрочем, при наличии проблем сообщения могут и пригодиться.



› **Протокол XMPP/Jabber широко распространен, и с ботом можно соединяться через разные клиенты, включая *Pidgin*.**

Даешь Disco

XMPP включает структуру модулей расширения под названием *Disco*. Они обогащают протокол другими типами сообщений, включая SIP (голосовые) и передачу файлов, и всеми мыслимыми возможностями, для которых применяется сеть «точка–точка». Так почему бы не прибавить функциональности и нашему боту – например, превратить его в хранилище файлов или чтеца новостей?

К счастью, Google расширил свою реализацию XMPP некоторыми дополнительными функциями, подробную информацию о которых вы найдете на сайте http://code.google.com/apis/talk/jep_extensions/extensions.html.

Создание соединения

После установки уровня отладки мы создали соединение (пароль тут подставьте ваш собственный) и получили ответный результат **tls,** который означает, что с сервером установлено безопасное соединение – вы можете также увидеть **tcp** для стандартного сетевого соединения, или пустую строку, если в соединении было отказано.

При малой прыткости ваших пальцев и в зависимости от вашего сервера, следующий шаг может не сработать: серверы Google ожидают от вас быстрой аутентификации соединения, в противном случае оно оборвется, оставив вас скрести в затылке и раздумывать, что за беда тут стряслась. Вам дается всего несколько секунд, и вы, возможно, захотите объединить следующие строки вместе:

```
>>> myclient.connect() ; myclient.auth(jid.
getNode(),'botpasswd')
'tls'
'sasl'
```

Результат **sasl** говорит о том, что соединение аутентифицировано с использованием SASL (Simple Authentification and Security Layer, Простая аутентификация и слой безопасности) посредством вашего пароля. Последний шаг установки соединения – объявление вашего статуса. Как вы знаете, в Google Chat и других сервисах Jabber существует несколько различных уровней статусов клиентов (например, Доступен или Занят). Кроме всего прочего, они используются сервером для составления списков доступных контактов, а в *Xmpppy* для этого существует специальный метод:

```
myclient.sendInitPresence()
```

Учтите, что некоторые серверы не дадут вам и шагу ступить, пока вы не укажете это правильным образом.



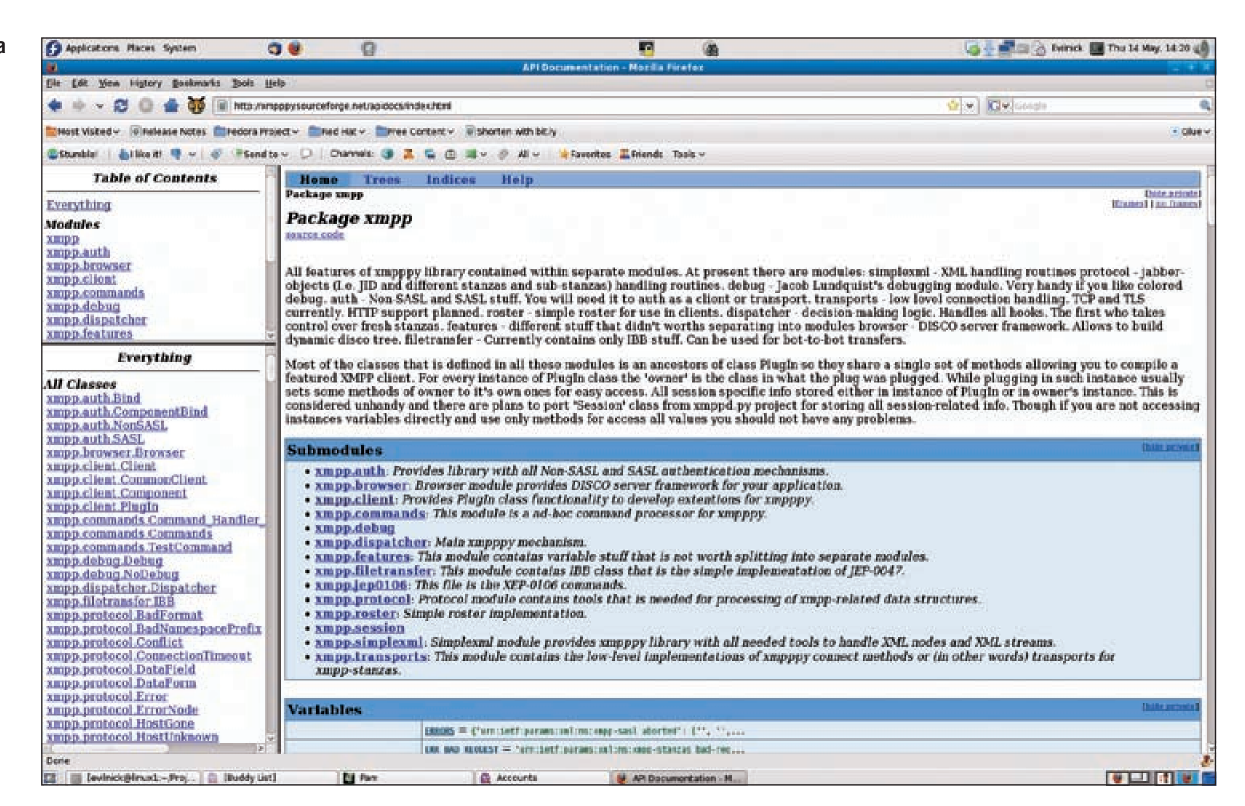
Безопасность вашего бота легко укрепить, ограничив ему прием команд только одним конкретным ID.

Послание в бутылке

Ура, есть контакт! Теперь, прежде чем заняться скучными вещами, давайте составим и отправим сообщение нашему оппоненту (хороший кандидат – ваш собственный адрес Google). Однако сначала нужно обзавестись адресом, куда требуется доставить сообщение, и его текстом. После этого создадим сообщение и отправим его

```
mymsg=xmpp.protocol.Message("evilbot.1@gmail.
```

»



› **Страница проекта *Xmpppy* не богата документацией, но справочник по API внятно объясняет, как работает модуль.**

```
com","hello", "chat")
myclient.send(mymsg)
```

Если вы справились с приглашением и разрешением на чат от учетной записи вашего бота, вы должны получить приятный сюрприз.

Одна штука может вас озадачить: а как установить свое статусное сообщение? Сервер отображает только статус и соответствующее сообщение, переданные клиентом.

На сервере Google это означает, что вы можете иметь статус Не на месте, Невидим, Отсутствую, Не беспокоить или На месте, а также в форме текстового сообщения. А вот и разгадка: ваш статус задается отправкой специального сообщения XMPP, которое идет прямо на сервер. Оно включает ваш статус и текст статусного сообщения. Не зная этого, самому догадаться сложно, но смысл тут есть: XMPP как раз предназначен для отправки сообщений, так зачем же заводить отдельные протоколы только для отправки сообщений про статус? Сервер отвечает за обновление статуса для всех, кто соединен с вами. Кстати говоря, протокол XMPP позволяет отправлять контактам явные сообщения о статусе, но, хотя тут и есть с чем порезвиться, обычно мы перекладываем эту заботу на сервер:

```
>>> presence = xmpp.Presence(status = "Ready!", show =
"chat", priority = '1')
>>> myclient.send(presence)
```

Модуль *Xmpppy* работает с сообщениями о статусе иначе, чем с обычными, потому что они используют другие аргументы, но механизм отправки тот же самый.

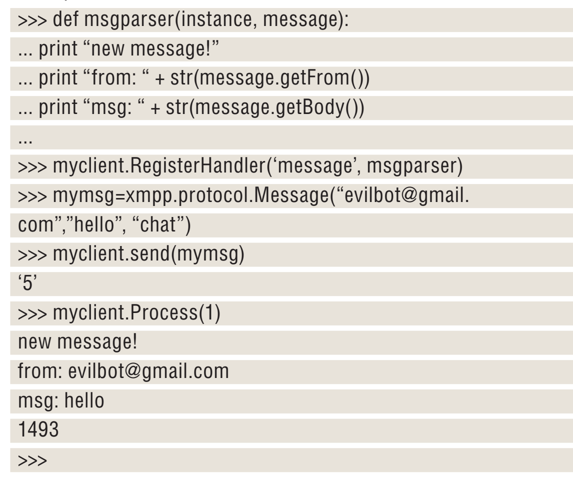
Почему Python?

Можно, естественно, писать скрипты и приложения для обработки web-данных на любом другом языке; так почему же мы выбрали Python, а не C#, например? На то есть веские причины. Python – язык простой и понятный, код на нем легко пишется и (что, вероятно, еще важнее) легко читается. Он обладает прекрасными возможностями для работы с текстом (которым по большей части и являются наши данные), он кросс-платформенный и снабжен бездной полезных библиотек для web-служб и протоколов. Используя Python и пару библиотек, можно мигом создать рабочее приложение или скрипт.

Прослушивание

От хорошего бота также нужно получать входящие сообщения. Среда *Xmpppy* принимает весточки, отправленные на созданный вами клиентский объект, и сохраняет их в стеке для обработки. А какой? *Xmpppy* использует концепцию обработчиков [handler]: сперва вы определяете функцию или метод, который будет действовать как получатель информации. Затем, при готовности обработать сообщения из стека, вы просто вызываете метод **Process()** объекта клиента. Звучит это сложнее, чем есть на самом деле – другими словами, вы сообщаете клиенту, где нарыть сообщений, затем велите ему пропустить их через функцию. Конечно, это можно сделать и через командную строку Python, но тогда будет сложнее: нужно написать функцию обработчика, а затем, обычным порядком, запустить бесконечный цикл для обработки сообщений по мере их прибытия.

Мы покажем соответствующий код в командной строке, но разумнее будет встроить его в собственный класс (как мы увидим далее). В данном примере, мы также отправим сообщение самим себе для тестирования, а вы можете сделать это через Gmail:



```
>>> def msgparser(instance, message):
... print "new message!"
... print "from: " + str(message.getFrom())
... print "msg: " + str(message.getBody())
...
>>> myclient.RegisterHandler('message', msgparser)
>>> mymsg=xmpp.protocol.Message("evilbot@gmail.
com","hello", "chat")
>>> myclient.send(mymsg)
'5'
>>> myclient.Process(1)
new message!
from: evilbot@gmail.com
msg: hello
1493
>>>
```

Как видите, анализирующая функция **msgparser()** проста. Она использует доступные методы **getFrom()** и **getBody()** во входящем

сообщении, преобразует результаты в строку и выдает на консоль. Для реального бота надо еще записать отправителя в переменную (чтобы ему отвечать), а может быть, разобрать сообщение глубже, чтобы сгенерировать ответ.

Помощь по Python

Если вы новичок в Python, но имеете опыт программирования на других языках, трудностей у вас возникнуть не должно: главное – помните, что надо правильно делать отступы. На основном сайте Python имеется масса документации по особенностям языка и синтаксису, а также по использованию стандартных модулей, поставляемых вместе с Python.

Командный голос

Для нашего простенького бота мы примем такой синтаксис: специальные команды будут начинаться с символа #. Получив сообщение, начинающееся с #, мы попытаемся сделать что-нибудь; а в противном случае вернем произвольный ответ из припасенного списка.

Добавление обработчика команд сделало бы код громоздким. Ради экономии места и умственных усилий применим трюк: используя конструкцию **eval** языка Python, мы создадим функцию, вызывающую метод в нашем классе с тем же именем, что и команда, которая была нам передана. Это, конечно, отчасти халтура, но экономит место и облегчает добавление команд – вам останется только определить новый метод. Для бота более продвинутого вы, возможно, захотите вставить механизм обработчиков, чтобы к экземплярам класса можно было добавлять свои команды, но ниже показано, как работает метод попроще (заметим, что данный код – это метод класса)

```
def messageparse(self, instance, message):
 m = message.getBody()
 sender=message.getFrom()
 if (m[0]=='#'):
  self.log('command received')
  #специальный случай
  #просто команда – подразумеваем, что
метод существует, как и id отправителя
  try:
   eval('self.'+m[1:]+'(sender)')
  except:
   self.client.send((xmpp.protocol.Message(sender, 'Sorry
Dave, I can\'t do that...')))
 else:
  #полагаю, надо сказать что-нибудь, из вежливости
  self.client.send((xmpp.protocol.Message(sender, random.
choice(self.responses))))
```

Как видите, сообщение проверяется на наличие символа #, и если он есть, мы создаем вызов метода по остатку строки, включая ID отправителя, и пытаемся выполнить его. Конструкции **try** и **execept** отлавливают исключения – например, несуществующий метод. Если команды нет, мы просто отправляем случайный ответ, выбрав его из списка **responses**. На самом деле, для этого нужно импортировать модуль **random**; **random.choice** случайным образом выбирает элемент из тех, что ему подсунули.

Обработчик команд может быть таким:

```
def uptime(self, sender):
 import subprocess
 p=subprocess.Popen(["uptime"], stdout=subprocess.PIPE)
 r=p.communicate()[0]
 self.client.send((xmpp.protocol.Message(sender,r)))
```

Объясним его, потому что он работает с запуском команд локально, на компьютере, где выполняется бот. Сначала идет определение, которое принимает экземпляр объекта **self** (это требование Python) и информацию об отправителе сообщения, добытую обработчиком. А потом мы импортируем **subprocess** из стандартных библиотек Python, для запуска локальной команды.

Кроме того, мы использовали метод под названием Popen (см. строку, начинающуюся с **p=subprocess.Popen)**, который задокументирован и объяснен по адресу http://docs.python.org/library/subprocess.html. Вкратце, мы передаем команду для выполнения и требуем, чтобы стандартный вывод отсылался в определенное место. Тогда мы можем связаться с этим выводом, используя метод **communicate** экземпляра Popen, и получить результат команды. Последняя строка запаковывает его и отправляет в виде чат-сообщения. Теперь вы можете делать запросы к вашему серверу через чат-клиент и заставлять его злодействовать!

Идем дальше

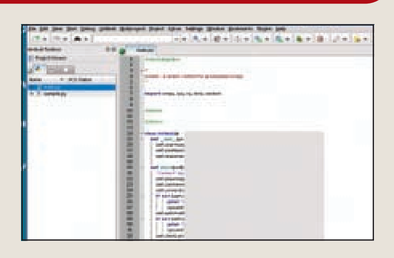
Полный код с комментариями для класса нашего чат-бота включен на LXFDVD. Если вы собираетесь запустить его, подставьте данные учетной записи вашего бота. А если вы импортируете его в качестве модуля, можете ввести данные об учетной записи и пароле просто через конструктор. Код нашего примера должен предоставить вам достаточно информации, поясняющей, как он работает.

Возможно, вам захочется добавить к классу собственные методы для задания других команд: автоматизировать какие-нибудь задания или обратиться к внешним ресурсам – например, создать систему напоминаний, использующую онлайн-календарь. Или, почему бы не приткнуть вашего чат-бота к сервису переводов – включить его в беседу и заставить переводить все, что вы скажете, на другой язык?

Чат-бот – всего лишь перевалочный пункт: легкий в использовании интерфейс, позволяющий обращаться к скрипту. Как вы трудоустроите своего чат-бота, зависит от вас, от того, на что вы его запрограммируете. Если придумаете другие способы его применения, пожалуйста, напишите нам; а если вы чего-то недопоняли – не забудьте просмотреть полный код на DVD. LXF

Редакторы Python

Python строг насчет синтаксиса. Это не так плохо, но чревато проблемами, если вы пишете код в неподходящем редакторе. Мы использовали *Vim и Kate* – оба имеют выделение синтаксиса и другие функции для облегчения задачи создания работающего скрипта на Python. Годится также *Eric*, IDE на базе Python, куда включено много других возможностей, специфичных для этого языка. Она доступна в большинстве дистрибутивов, и ее можно скачать с сайта http://eric-ide.python-projects.org/eric4-download.html.



› Используйте редактор, умеющий подсвечивать синтаксис для Python, или специальный, из IDE *Eric*.

Версии Python

Python 3.0 уже доступен, но поскольку его код несовместим с прежними версиями, многие дистрибутивы по умолчанию все еще используют (и будут использовать в обозримом будущем) версии 2.x. Поэтому весь наш код в этой серии уроков совместим с 2.x, для удобства большинства читателей.

» **Через месяц** Twitter… Научим Python слать 140-символьные сообщения.

Inotify: Действия

Создание, изменение и удаление файлов и каталогов – это вещи, происходящие в вашей системе по тысяче раз на дню. Их можно отследить и применить для вашего удобства, утверждает **Николай Кузнецов**.



Наш эксперт

Николай Кузнецов
Ведет курсы по Linux уже более пяти лет. Из обученных им за это время получился бы ударный полк линуксоидов.

Для начала – немного теории. *Inotify* – это подсистема ядра Linux, позволяющая отслеживать события в файловой системе. Причем происходит это в так называемом «реактивном режиме» (то есть по событию), в отличие от режима «активного» – постоянного сканирования изменений. Это предоставляет нам достаточно эффективный механизм для самых различных нужд. И хоть появилась эта функция сравнительно недавно (начиная с ядра версии 2.6.13-rc3), ее рождение повлекло за собой значительные и не очень улучшения в различных программах, таких как: *Beagle* (система индексации и быстрого поиска файлов на рабочем столе), *auditd* (служба аудита), *Brasero* (программа записи CD/DVD в GNOME), *tailf* (эффективный аналог команды **tail -f**), *udevd* (подсистема работы с устройствами) и многие другие. Мы же продемонстрируем возможности *Inotify* на примере двух утилит: *iwatch* и *incron*. На данном уроке будет использоваться SUSE 11, но все действия будут аналогичными и в других дистрибутивах. Если же вы предпочтете SUSE, то сможете найти все указанные пакеты при помощи поиска на сайте http://software.opensuse.org/search.

iПосмотрим: iwatch

С помощью *iwatch* можно легко наблюдать, какие изменения происходят в определенной части дерева каталогов файловой системы. Для примера, запустим следующую команду:

```
# iwatch -v -r /etc/
```

Она будет подробно (**-v**) сообщать обо всем, что происходит в директории **/etc** и во всех ее поддиректориях (**-r**). Теперь можно открыть вашу любимую графическую программу для настройки чего-нибудь (в SUSE попробуйте *YaST*, в Fedora – утилиты *system-config-...*) и посмотреть, какие изменения она вносит в общесистемные конфигурационные файлы.

Давайте переключимся на другой терминал и выполним:

```
# touch /etc/issue
```

В результате на консоли с *iwatch* можно будет увидеть следующее:

```
[15/Jun/2009 14:08:02] IN_CLOSE_WRITE /etc/issue
[15/Jun/2009 14:08:02] * /etc/issue is closed
```

Рассмотрим более сложный пример: разберемся, что происходит при создании нового пользователя командой

```
# useradd testuser
```

Вывод *iwatch* здесь будет куда более многословным:

```
[15/Jun/2009 14:10:06] IN_CREATE /etc/passwd.tmpZRwKCR
[15/Jun/2009 14:10:06] IN_CLOSE_WRITE /etc/passwd.tmpZRwKCR
[15/Jun/2009 14:10:06] IN_DELETE /etc/passwd.old
[15/Jun/2009 14:10:06] * /etc/passwd.old is deleted
[15/Jun/2009 14:10:06] IN_CREATE /etc/passwd.old
[15/Jun/2009 14:10:06] IN_MOVED_FROM /etc/passwd.tmpZRwKCR
[15/Jun/2009 14:10:06] IN_MOVED_TO /etc/passwd
[15/Jun/2009 14:10:06] * /etc/passwd.tmpZRwKCR is moved to /etc/passwd
[15/Jun/2009 14:10:06] IN_CREATE /etc/shadow.tmpCsjybQ
[15/Jun/2009 14:10:06] IN_CLOSE_WRITE /etc/shadow.tmpCsjybQ
[15/Jun/2009 14:10:06] IN_DELETE /etc/shadow.old
[15/Jun/2009 14:10:06] * /etc/shadow.old is deleted
[15/Jun/2009 14:10:06] IN_CREATE /etc/shadow.old
[15/Jun/2009 14:10:06] IN_MOVED_FROM /etc/shadow.tmpCsjybQ
[15/Jun/2009 14:10:06] IN_MOVED_TO /etc/shadow
[15/Jun/2009 14:10:06] * /etc/shadow.tmpCsjybQ is moved to /etc/shadow
[15/Jun/2009 14:10:06] IN_CREATE /etc/group.tmpfr7KLO
[15/Jun/2009 14:10:06] IN_CLOSE_WRITE /etc/group.tmpfr7KLO
[15/Jun/2009 14:10:06] IN_DELETE /etc/group.old
[15/Jun/2009 14:10:06] * /etc/group.old is deleted
[15/Jun/2009 14:10:06] IN_CREATE /etc/group.old
[15/Jun/2009 14:10:06] IN_MOVED_FROM /etc/group.tmpfr7KLO
[15/Jun/2009 14:10:06] IN_MOVED_TO /etc/group
[15/Jun/2009 14:10:06] * /etc/group.tmpfr7KLO is moved to /etc/group
[15/Jun/2009 14:10:06] IN_CREATE /etc/group.tmpwsLtmN
[15/Jun/2009 14:10:06] IN_CLOSE_WRITE /etc/group.tmpwsLtmN
[15/Jun/2009 14:10:06] IN_DELETE /etc/group.old
[15/Jun/2009 14:10:06] * /etc/group.old is deleted
[15/Jun/2009 14:10:06] IN_CREATE /etc/group.old
[15/Jun/2009 14:10:06] IN_MOVED_FROM /etc/group.tmpwsLtmN
[15/Jun/2009 14:10:06] IN_MOVED_TO /etc/group
[15/Jun/2009 14:10:06] * /etc/group.tmpwsLtmN is moved to /etc/group
[15/Jun/2009 14:10:06] IN_CLOSE_WRITE /etc/.pwd.lock
[15/Jun/2009 14:10:06] * /etc/.pwd.lock is closed
```

В листинге видны следующие inotify-события:

- » **IN_CREATE** – был создан файл или директория;
- » **IN_MOVED_FROM\IN_MOVED_TO** – файл перемещался (откуда и куда),
- » **IN_DELETE** – файл был удален,
- » **IN_CLOSE_WRITE** – файл был закрыт после открытия в режиме для записи.

Из вывода *iwatch* становится ясно, какие файлы создавались/перемещались/удалялись во время выполнения команды **useradd**. В целом, картина известная: изменяются файлы **/etc/passwd**, **/etc/shadow** и **/etc/group**, а их старые копии сохраняются в **/etc/passwd.old**, **/etc/shadow.old** и **/etc/group.old**, соответственно. А вот про существование файла **/etc/.pwd.lock** я, например, не знал.

Кстати, обратите внимание, как происходит изменение файлов. Содержимое не перезаписывается: вместо этого создается новый временный файл, в который вносятся все необходимые строки, после чего он переименовывается. Помните споры вокруг проблем с отложенной записью в ext4? Отрадно понимать,



› **Мы использовали сценарий на *Bash*, но подойдет и любой другой язык программирования: Python, Perl и даже C.**

по событиям

что **useradd** (как, впрочем, и все грамотно написанные приложения Unix) этой проблеме не подвержен.

Получив представление о событиях *Inotify*, приступим к настройке службы *incron*.

inПопланируем: incron

Что же такое *incron*? Это – вариация службы *Cron*, выполняющая задания по событиям *Inotify*, а не по таймеру. Давайте реализуем при его помощи автоматическую сортировку по типам для файлов, сбрасываемых на рабочий стол пользователя testuser: музыка – в один каталог, фотоснимки – в другой, и так далее.

Для этого создаем правило *incron*, запустив следующую команду:

```
# incrontab -e
```

и набрав такую строку:

```
/home/testuser/Desktop IN_CREATE /usr/local/bin/Download_
Sort.sh $#
```

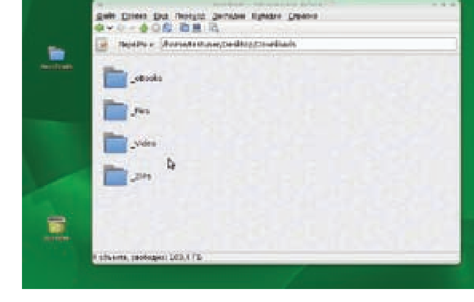
Таким образом, мы будем отслеживать событие In_CREATE (появление нового файла или директории) в каталоге **/home/testuser/Desktop**, и в случае, если оно произойдет, запускать скрипт **/usr/local/bin/Download_Sort.sh**, передавая ему в качестве аргумента имя созданного файла (оно, согласно документации *incron*, хранится в переменной **$#**).

Далее, создаем сам сценарий **/usr/local/bin/Download_Sort.sh**:

```
#!/bin/bash
DIR=/home/testuser/Desktop
{
echo "File: $i"
case "$1" in
      *.avi|*.flv|*.vob)
            mv -v "$DIR/$1" $DIR/Downloads/_Video
      ;;
      *.png|*.gif|*.jpg)
            mv -v "$DIR/$1" $DIR/Downloads/_Pics
      ;;
      *.pdf|*.djvu)
            mv -v "$DIR/$1" $DIR/Downloads/_eBooks
      ;;
      *.zip)
                        newDIR="$DIR/Downloads/_ZIPs/${1%.*}"
            mkdir "$newDIR"
                        sleep 20; while [ -f "$DIR/$1.part" ]; do
sleep 5; done
            unzip "$DIR/$1" -d "$newDIR/"
            mv -v "$DIR/$1" "$newDIR/"
            chown -R testuser "$newDIR"
      ;;
      *)
                        echo " - None"
      ;;
esac
} &> /tmp/Download_Sort.log
```

Принцип его работы очень прост: файлы распознаются по расширениям (при желании сделать скрипт более «интеллектуальным» можно использовать команду **file**). Видеоролики перемещаются в папку **_Video**, картинки – в **_Pics**, документы PDF и DjVu – в **_eBooks**, Zip-архивы распаковываются в **_ZIPs**, файлы неизвестных форматов остаются на рабочем столе. Кроме этого, результат и ошибки обработки последнего файла сохраняются в **/tmp/Download_Sort.log** для проверки и тестирования работы скрипта. Сообщения службы *incrond* можно также просмотреть в системном журнале (**/var/log/messages** или аналогичном). Они имеют стандартный вид:

```
Jun 15 14:36:11 hostname incrond[4728]: (root) CMD (/usr/local/
bin/Download_Sort.sh Pinguins.zip)
```



› **Аккуратный рабочий стол — не главное в этом примере. Главное — возможность автоматизировать рутинные задачи.**

Теперь создадим **/home/testuser/Desktop/Downloads** и соответствующие поддиректории для разных типов файлов:

```
mkdir -p /home/testuser/Desktop/
Downloads/_{Video,Pics,eBooks,ZIPs}
```

Установим права, необходимые для запуска скрипта и для нормальной работы пользователя с его каталогами:

```
chmod a+rx /usr/local/bin/Download_Sort.sh
chown -R testuser /home/testuser/Desktop/Downloads
```

И, наконец, запустим службу, которая будет отслеживать для нас события:

```
/etc/init.d/incron start
```

Ее можно добавить и в автозапуск, если вы хотите, чтобы она самостоятельно стартовала после перезагрузки компьютера:

```
chkconfig incron on
```

Вот и всё: теперь можно сбрасывать на рабочий стол (в каталог **/home/testuser/Desktop**) различные файлы и смотреть, что с ними происходит.

Для наглядности, здесь приведен достаточно простой вариант скрипта. Желающие могут написать свой или расширить предложенный: например, добавив действия для RPM (установка или прописывание в локальный репозиторий), JAR (перенос на смартфон), EXE (проверка антивирусом), WAR (их – Tomcat'у), FLV (перекодирование в AVI), MP3 (в Ogg) и т.д., и т.п.

Кроме того, с помощью *incron* можно легко реализовать следующие идеи:

» Забыть про перезапуск сервисов после правки файлов конфигурации– этим займется *incron*, следящий за их изменением.

» Реализовать свою достаточно гибкую службу квот и аудита.

» Автоматически добавлять в свою web-галерею папку с картинками при копировании, например, в **/srv/www/htdocs/images**.

Уверен, вы сможете придумать и свои собственные способы для удобного использования данного инструмента. LXF

Скорая помощь

Вместо редактора по умолчанию (*Vim*) правила *incron* можно вводить при помощи любого другого, предварительно выполнив команду вроде **export EDITOR=gedit** или **export EDITOR=mcedit**., то есть установив переменную окружения **$EDITOR**.

Выбор есть всегда

Разумеется, ваше взаимодействие с *Inotify* не ограничивается *iwatch* с *incron*: в Linux есть весь необходимый инструментарий. Программистам на C обычно достаточно подключить заголовочный файл **<linux/inotify.h>**, разработчикам на Perl и Python пригодятся пакеты *python-pyinotify* и *perl-Linux-Inotify2*, web-программистам – *php5-inotify*... модуль *Inotify* есть даже в *Compiz*! В скриптах удобно пользоваться утилитой *inotifywait* из пакета *inotify-tools* – как следует из названия, она позволяет приостановить сценарий в ожидании какого-либо события файловой системы.

Linux: Сетевой

Изобретать велосипед – задача неблагодарная, а вот разбирая готовый велосипед, можно узнать много нового и полезного. **Артем Коротченко** напишет для вас утилиту *Ping*.



Наш эксперт

Артем Коротченко
Интересуется низкоуровневым программированием, встраиваемыми системами, а также приспособлением для них свободных ОС.

Давайте вспомним самое начало истории нашей любимой операционной системы. В 1991 году один простой, но весьма смышленый финский студент опубликовал исходные коды первой версии ядра своей собственной ОС. Тогда она была еще неработоспособной и не особо полезной на практике, однако эксперимент по созданию первого свободного Unix заинтересовал определенные круги программистов.

Удивительно, что все началось не с закрытой разработки какой-нибудь коммерческой организации, а с сообщения в публичной телеконференции Usenet. Само по себе это уже является нонсенсом в индустрии программного обеспечения начала девяностых. Однако я назвал Linux сетевым проектом даже не потому, что он – дитя Интернета. К настоящему времени Linux прошел огромный путь, но до сих пор является плодом труда тысяч хакеров-энтузиастов. До сих пор его разработка не централизована (не имеет лидера), из чего следует второй смысл утверждения, вынесенного в заголовок: Linux – сетевой, в смысле децентрализованный, проект.

Итак, Сеть объединила самых талантливых людей со всего мира в работе над одним общим делом, с великой и благородной целью. Естественно, первоклассная поддержка Интернета и сетевого взаимодействия в такой системе должна была стать одной из важнейших ее особенностей. Понять, насколько верно это предположение, можно, оценив лишь тот факт, что многие (если не большинство) серверы в Интернете работают под управлением Linux.

> «Обмен информацией между хостами — сложный процесс.»

Сетевая суть Linux отразилось и на писавшемся для него прикладном ПО. Даже те игровые серверы, клиентские части которых рассчитаны на Windows, в большинстве своем написаны для Linux. Нет сомнений в изобилии серверов Web, почты, FTP, DNS... Количество клиентских приложений и вовсе неисчислимо.

Рожденный в Сети, Linux моментально обзавелся собственной реализацией интернет-протоколов. Освоить программирование в Linux – это значит научиться писать для него сетевое ПО.

Сетевое взаимодействие

Самая популярная сетевая программа – это, пожалуй, web-клиент. Поэтому я сначала думал написать маленький web-браузер. Но в таком случае программировать бы пришлось только HTTP-протокол, не спускаясь на нижние уровни.

Поэтому я остановился на всем известной утилите *ping*. Она легко реализуема, проста по сути и основана на протоколе низкого уровня (ICMP), что делает ее идеальной для начала разбирательства в сетевом программировании. Но что такое уровни протоколов? Что такое протокол? Глубоко ознакомиться с этой темой можно в Интернете (например, почитать документы RFC); я лишь затрону самое необходимое.

Обмен информацией между двумя хостами – сложный процесс. Международная организация по стандартизации применяет для его описания семь универсальных уровней OSI. Для нашего случая (TCP/IP) они объединены в четыре. Каждому уровню соответствуют протоколы – языки, благодаря которым узлы сети понимают друг друга.

В самом простом смысле взаимодействие двух систем представляет собой чередование электрических импульсов; для Wi-Fi — передачу модулированного радиосигнала. Но могут быть даже механические/акустические вибрации: во всех трех случаях речь идет об уровне доступа к сети (Network Interface Layer). О связанной с ним информации говорится как о наборе кадров; в качестве протоколов здесь фигурируют Ethernet, PPP.

Физические сигналы несут в себе некие потоки данных (электронное письмо, закачка файлов по FTP, онлайн-телевидение), суть которых второму хосту разобрать пока что нереально. А главное, неизвестно, кому они адресованы. Ведь очевидно, что между обменивающимися системами всегда будут единицы, десятки и сотни посредников; не будет их только в соединении напрямую.

Поэтому кадры сопровождаются адресами: исходным и получателя. Тогда говорится уже о втором, межсетевом уровне (Internet Layer), задача которого заключается в маршрутизации передаваемых данных до точки назначения. Об информационных потоках теперь говорится не как о последовательности битов, а как о датаграммах. Протокол, призванный распознавать адреса в любых датаграммах – IP (по правде сказать, есть еще MAC-адреса в Ethernet-заголовке, но сегодня они нас не интересуют). ICMP-протокол (диагностические сообщения, ошибки) также относится к межсетевому уровню.

Далее производится проверка. Если адрес стороны, получившей пакет, не совпадает с адресом назначения, система передает его дальше по сети. Таким образом, выше второго уровня чужие пакеты никогда не поднимаются. Если адреса совпадают, то можно уже думать, какому именно локальному сервису предназначается содержимое пакета. Для этого он поднимается на транспортный уровень (Host-to-Host Layer). По IP-заголовку вычисляется тип пакета (TCP или UDP), и соответствующая подсистема читает из этих данных свою ключевую информацию – порты отправляющей и принимающей стороны. TCP-заголовки содержат также флаги-параметры (запрос разрешения на установление сеанса, подтверждение сеанса, завершение и другие). В UDP-взаимодействии не происходит установления стабильного сеанса, поэтому этот протокол является более быстродействующим, но менее надежным (и от того менее популярным) аналогом TCP. В общем, связка TCP и IP – это основа функционирования Интернета.

Допустим, контрольные суммы совпали, никаких проблем не возникло. Теперь данными занимается подсистема прикладного уровня (Application Layer). Пакеты с 80-м TCP-портом будут отправлены Web-серверу, с 21-го их получит FTP, 25-м займется SMTP. Конечно, это произойдет, если соответствующие демоны запущены, работают и не перенастроены на нестандартные порты. Иначе говорится, что порт закрыт, и тогда отправленные ему

проект



пакеты отбрасываются. (Это для сервера, клиентской стороне слушать порты обычно не нужно.)

Поскольку каждый уровень перед отправкой данных наверх убирает свой заголовок, ни одна из сетевых подсистем не знает предыстории пакетов. Прикладному уровню достаются чистые данные. Например, все, что увидит Web-сервер – это запрос на выдачу некой страницы, а Jabber-сервер получит XML-код вида:

```
<iq type='get' id='auth_некий md5-хэш'><query xmlns='jabber:iq:auth'>
<username>имя пользователя, под которым клиент собирается войти в систему</username>
</query></iq>
```

В соответствии с протоколом прикладного уровня Jabber (правильнее сказать, XMPP) он сгенерирует XML-ответ с просьбой ввести пароль. И теперь пойдет обратный процесс – формирование пакета. Это сообщение будет спускаться с верхнего уровня к нижнему, снабжаясь заголовками с IP-адресами, контрольными суммами и прочими атрибутами, а затем уйдет в Сеть.

Приступим к делу

Вся суть *ping* сводится к тому, чтобы посылать на адрес проверяемого хоста запросы ICMP Echo-Request и получать (или не получать) от него ответы ICMP Echo-Reply. Кроме того, утилита должна учитывать время задержки пакетов в сети.

Наша реализация будет не хуже оригинала, за тем лишь исключением, что вместо десятков входных параметров она сумеет понимать лишь самые важные: **--help**, **--version**, **-c** и, конечно, IP-адрес исследуемой системы. Аргумент «**c**» отвечает за количество запросов, которые нужно отослать. Если он не будет задан, приложение будет работать до тех пор, пока его не завершат вручную. Обработка аргументов целиком реализуется с помощью функции getopt_long(), подробнее о которой вы можете прочитать в номере **LXF112**.

Помимо основных заголовков, нам нужны библиотеки сетевых функций и структур пакетов:

```
#include <netdb.h>
#include <netinet/ip.h>
#include <netinet/ip_icmp.h>
```

Теперь можно создать сокет:

```
int socket(int domain, int type, int protocol);
```

Первым аргументом этой функции обычно выбирается константа PF_INET, означающая, что мы желаем работать с протоколами IPv4.

Чтобы получить доступ к транспортному уровню, вторым аргументом указывается SOCK_STREAM для соединений TCP и SOCK_DGRAM для UDP. Третий аргумент всегда равен 0. Но чтобы формировать ICMP-пакеты, нам нужен межсетевой уровень, поэтому будем использовать так называемые «сырые» сокеты:

```
sp = socket(PF_INET, SOCK_RAW, IPPROTO_ICMP);
```

Можно идти дальше и задавать сокеты для доступа к заголовкам физического уровня (например, для работы с ARP-пакетами, выявляющими соответствие между IP- и Ethernet-адресами), но нам пока это не требуется.

Разрешим широковещательные сообщения и увеличим размер приемного буфера:

```
int on = 1, size = 61440;
…
setsockopt(sp, SOL_SOCKET, SO_BROADCAST, &on, sizeof(on));
setsockopt(sp, SOL_SOCKET, SO_RCVBUF, &size, sizeof(size));
```

Заданный во входных параметрах IP (хост) представляет собой строку, массив переменных типа char. Но этого мало, и нужно использовать специальные адресные структуры servaddr, которые будут содержать более полную информацию об адресате.

```
struct hostent *hp;
struct sockaddr_out servaddr;
...
/* iparg – указатель на аргумент с адресом (arv[]) */
hp = gethostbyname(iparg);
...
/* Обнуляем структуру */
bzero(&servaddr_out, sizeof(servaddr_out));
servaddr_out.sin_family = AF_INET;
servaddr_out.sin_addr = *((struct in_addr *) hp->h_addr);
```

Для получения пакетов нам потом понадобится еще одна такая структура – sockaddr_in.

В нашей программе будет два типа сигналов: информирующий о необходимости послать новый пакет (генерируется каждую секунду) и завершающий (например, по Ctrl+C). Назначим им функции-обработчики, зададим таймер:

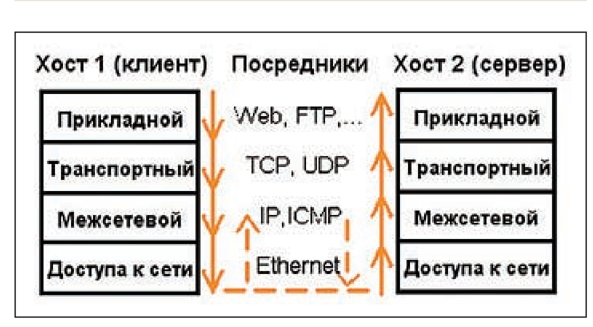
```
struct itimerval tval;
...
/* Обнуляем значение интервала */
timerclear(&tval.it_interval);
/* Устанавливаем значение интервала - 1 секунду */
tval.it_interval.tv_sec = 1;
/* Обнуляем значение времени срабатывания */
timerclear(&tval.it_value);
/* Устанавливаем его в 1 микросекунду */
tval.it_value.tv_usec = 1;
/* Связываем сигнал таймера с функцией-обработчиком */
(void) signal(SIGALRM, handler);
/* Связываем сигнал завершения с функцией-обработчиком */
(void) signal(SIGINT, handler);
/* Запускаем таймер */
(void) setitimer(ITIMER_REAL, &tval, NULL);
```

Прием и передача

Это были подготовительные действия. Теперь нужно отправлять и принимать пакеты. Первое возлагается на обработчик handler():

```
void handler(int signo)
{
  int losscount;
  struct icmp *icmp;
```

»



› Наш клиент отправляет что-то серверу (четырехуровневая модель TCP/IP).

```
 char sndbuf[BUFSIZE];
 if(signo == SIGINT) {
  /* … Вывести результаты - код опущен … */
  exit(0);
 }
 if(signo == SIGALRM) {
  icmp = (struct icmp *) sndbuf;
  icmp->icmp_type = ICMP_ECHO;
  icmp->icmp_code = 0;
  icmp->icmp_id = pid;
  icmp->icmp_seq = ++sntcount;
  gettimeofday((struct timeval *) icmp->icmp_data, NULL);
  icmp->icmp_cksum = in_cksum((unsigned short *) icmp, 64);
  if(sendto(sp,sndbuf,64,0,(struct sockaddr*)&servaddr_out,
sizeof(servaddr_out))<0){
   fprintf(stderr, "%s: sendto failed: %s\n", myname,
strerror(errno));
   exit(1);
  }
  fflush(stdout);
 }
}
```

Как видно, вторая часть функции занимается сигналами таймера. Каждую секунду некая область оперативной памяти компьютера объявляется буфером sndbuf, и в его бессмысленном наборе байтов происходит формирование пакета. С помощью указателя последовательно заполняются поля ICMP-заголовка:

» Тип и код выбираются такими, чтобы полученный пакет представлял собой именно *Echo Request*.

» Поле id выставляется по возвращаемому значению функции getpid() (это нужно, чтобы потом игнорировать сообщения Echo Reply, предназначенные другим процессам).

» sequence – порядковый номер отправляемого пакета.

» Контрольная сумма рассчитывается по алгоритму из RFC с использованием популярной реализации in_cksum().

В данные пакета записывается текущее время. Поскольку в ответах они будут дублироваться, мы сможем определять время отклика хоста. Нам не нужен доступ к IP-заголовку, а потому его заполнение доверяется системе.

Теперь, когда имеется открытый сокет, заполненная адресная структура и подготовленный пакет, функция sendto() отправляет его в Сеть.

Прием сообщений производится в бесконечном цикле main():

```
int rcvlen;
int servaddr_in_len = sizeof(servaddr_in);
/* Длины заголовков */
int ip_len, icmp_len;
/* Буфер для полученного пакета */
char rcvbuf[BUFSIZE];
/* Время оборота пакета в миллисекундах */
float time;
/* Указатель на IP-заголовок */
struct ip *ip;
/* Указатель на ICMP-заголовок */
struct icmp *icmp;
/* Структура времени получения пакетов */
struct timeval trcvbuf;
/* Указатели на структуры времени получения и отправки
пакетов */
struct timeval *trcv, *tsnd;
struct itimerval tval;
...
trcv = &trcvbuf;
/* Пакет начинается с IP-заголовка */
ip = (struct ip *) rcvbuf;
while(1) {
  /* Получаем пакет */
  rcvlen = recvfrom(sp, rcvbuf, sizeof(rcvbuf), 0, (struct sockaddr
*)&servaddr_in, &servaddr_in_len);
  if(rcvlen < 0) {
   if(errno == EINTR)
    continue;
   fprintf(stderr, "%s: recvfrom() failed: %s\n", myname,
strerror(errno));
   return 1;
  }
  ip_len = ip->ip_hl << 2;
  icmp = (struct icmp *) (rcvbuf + ip_len); // ICMP-заголовок идет
после IP-заголовка
  icmp_len = rcvlen – ip_len;
 /* Продолжаем, только если получили ICMP Echo Reply пакет,
который предназначается
    именно нашей программе */
  if(icmp->icmp_type == ICMP_ECHOREPLY && icmp->icmp_id ==
pid) {
   rcvcount++;
   tsnd = (struct timeval *) icmp->icmp_data;
   /* Посчитаем время оборота пакета путем вычитания вре-
мен получения
     и отправки пакетов. Секунды и микросекунды можно пе-
ревести
     в миллисекунды и сложить. */
   gettimeofday(trcv, NULL);
   time = (trcv->tv_sec-tsnd->tv_sec)*1000+(trcv->tv_usec-tsnd-
>tv_usec)/(float)1000;
   /* ... Вывод результатов на экран – код опущен ... */
   /* Если обработано заданное количество пакетов, заверша-
ем программу */
   if(sntcount == c && c)
    handler(SIGINT);
  }
}
```

Тут перед нами стоит противоположная задача. В буфер rcvbuf записываются принимаемые из Сети данные, и нужно выделить в нем структуры сетевых заголовков. Буфер начинается с IP-заголовка, одно из полей которого характеризует его размер. Обратившись к этому полю, легко определить, где кончается часть IP и начинается заголовок ICMP. Длина последнего необходима для вывода строки результатов, определяется она еще проще – до конца буфера.

С определением начала ICMP-заголовка нам становятся доступны его поля, которые мы сразу же и проверяем. Если все в порядке – получен пакет Echo Reply, притом предназначающийся нашему процессу – значит, счетчик полученных ответов можно увеличить на единицу. Обновляем статистику, и затем выводим промежуточные результаты. Все! Полные исходные коды ищите на DVD.

А если проще?

Написание сетевых приложений можно упростить, воспользовавшись популярными библиотеками *libpcap* и *libnet* (доступными, кстати, не только в Linux и Unix). Первая предоставляет простой интерфейс для обработки входящего трафика, вторая предназначена для создания пакетов перед их последующей отправкой.

У такого подхода есть свои плюсы и минусы. С одной стороны, мы теряем абсолютную свободу в программировании сети и вынуждаем себя таскать эти библиотеки вместе с приложением, с другой – значительно облегчаем свою жизнь и уменьшаем вероятность допустить ошибку в коде.

Что касается *ping*, то пользы от переписывания этой утилиты «простым путем» практически нет. *libpcap* в цикле main() обрабатывал бы входящие эхо-ответы. Ради интереса посмотрим, как *libnet* сформирует нам эхо-запрос. Добавим соответствующий заголовок:

```
#include <libnet.h>
```

Все последующие изменения касаются обработчика handler(), ведь именно он занимается отправлением пакетов. С помощью функции libnet_init() производим инициализацию сеанса:

```
libnet_t *l;
l = libnet_init(LIBNET_RAW4, NULL, errbuf);
if(l == NULL)
{
 fprintf(stderr, "%s: libnet_init() failed: %s\n", myname, errbuf);
 exit(1);
}
```

Библиотека позволяет работать со всеми сетевыми уровнями и создавать любые заголовки. Мы просто задаем нужные в тэгах протоколов и заполняем их поля:

```
libnet_ptag_t ip, icmp;
u_char data[BUFSIZE]; // Буфер поля данных
char errbuf[LIBNET_ERRBUF_SIZE]; // Буфер ошибок
gettimeofday((struct timeval *) &data, NULL); // Записываем время в данные
icmp = libnet_build_icmpv4_echo(ICMP_ECHO,   /* Тип */
                            0,        /* Код */
                            0,        /* Контрольная сумма */
                            pid,       /* ID */
                            ++sntcount, /* Номер очереди */
                            data,      /* Указатель на данные */
                            56,        /* Длина данных */
                            l,        /* Указатель на libnet_t */
                            icmp);
if(icmp == -1)
{
 fprintf(stderr, "%s: can't build ICMP header: %s\n", myname,
libnet_geterror(l));
 exit(1);
}
```

Как видно, выгода по сравнению с предыдущим подходом все же есть: теперь нам не нужно считать контрольную сумму, *libnet* займется ею сам. Мы также доверим ему заполнение IP-заголовка, самостоятельно указав лишь IP хоста.

```
ip4 = libnet_autobuild_ipv4(LIBNET_IPV4_H + LIBNET_ICMPV4_
ECHO_H + 56, /* Длина */
                IPPROTO_ICMP, /* Протокол */
                        destaddr,    /* IP
назначения */
                        l);          /* Указа-
тель на libnet_t */
if(ip4 == -1)
{
 fprintf(stderr, "%s: can't build IP header: %s\n", myname,
libnet_geterror(l));
 exit(1);
}
```

Кстати, IP-адрес назначения тоже обрабатывается из аргумента немного легче, без заполнения структуры hostent (а потом еще и sockaddr_in – как это было раньше).

```
u_long destaddr;
if((destaddr = libnet_name2addr4(l, iparg, LIBNET_RESOLVE)) ==
-1)
{/* Выдать ошибку */}
Пакет готов, его можно отправить, а сеанс завершить.
if(libnet_write(l) == -1) {/* Выдать ошибку */}
libnet_destroy(l);
```

Прослушивание в Сети

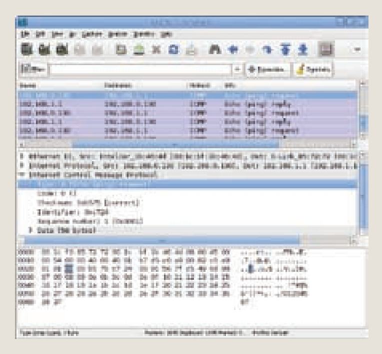
Можно ли посмотреть внутреннее устройство пакетов, «потрогать» их? Да, можно анализировать весь сетевой трафик, воспользовавшись программой-сниффером. Таким образом вы не только проверите все, о чем рассказывается в этой статье, но, возможно, к вам придет внезапное озарение вместе с глубоким пониманием основ Интернета.

Вероятно, в вашей системе уже есть *tcpdump* и *netstat* — классические утилиты Unix. Последняя, хоть и не является анализатором трафика, выдаст массу полезной информации обо всех входящих и исходящих соединениях. Если для вас важен GUI, следует обратить внимание на *ettercap* и особенно на *Wireshark* (ранее называвшийся *Ethereal*).

Давайте посмотрим на него поближе. Установить программу можно через менеджер пакетов вашего дистрибутива. Откройте ее, настройте на прослушивание нужного интерфейса (в моем случае это *wlan0*). Запустите наш *ping* – и, как ожидалось, в окне перехваченного трафика можно будет увидеть череду пакетов ICMP Echo-Request и ICMP Echo-Reply.

В нижнем окошке показывается некая последовательность байтов – это шестнадцатиричное представление отправленного эхо-запроса; в среднем – она разбирается по протоколам. Первым идет Ethernet-заголовок с его MAC-адресами и указанием на то, что это IP-пакет. Как вы помните, мы не трогали физический уровень. Далее идет IP-заголовок, для которого мы выбирали лишь IP-адрес назначения. Последующий ICMP-заголовок мы заполняли самостоятельно: тип, код, контрольная сумма, ID, Sequence. Знакомо? Если разобраться с последними 56 байтами собственно данных, увидим, что в них содержится системное время – мы определяли его функцией gettimeofday().

Более подробную информацию о *Wireshark* ищите на с. 42.



› ***Wireshark*** **может разобрать отправленный нашей программой пакет «по косточкам».**

Проверка боем

Скомпилируем утилиту. Никаких специальных ключей (если, конечно, вы не используете *libpcap* или *libnet*) для этого не потребуется. Просто наберите:

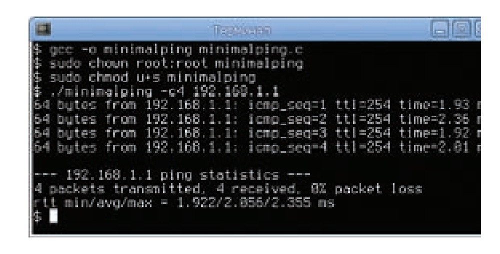
```
gcc -o minimalping minimalping.c
```

Прежде чем выполнить полученный файл, запомните: создавать raw-сокеты могут только приложения, запущенные привилегированным пользователем. Чтобы все остальные смогли пользоваться программой, необходимо выбрать ей владельца *root*, а также установить для нее SUID- или SGID-бит. Теперь можно сравнить результаты работы с оригинальным *ping* (см. рисунок).

Ну вот, а прошлая версия *ping* не отличалась совсем! Дело в том, что я перешел на Debian Lenny, и теперь для достижения абсолютной идентичности придется вновь править код: добавить выходные параметры *mdev* и *time*. Оставляю вам это в качестве домашнего задания.

Мы затронули лишь частичку удивительного мира. Программирование низких уровней стека TCP/IP открывает в нем безграничные возможности: черные ходы, сканеры, черви, снифферы, или, по другую сторону баррикады, honeypot-системы, брандмауэры и межсетевые экраны. С, особенно в связке с Linux, дает доступ к этим уровням. **LXF**

› **Штатная и разработанная нами утилиты *ping*: найдите одно отличие.**



Кодируем: Цвета

В очередном выпуске сборника советов для программистов **Андрей Боровский** затронет выполнение программ с правами root и управляющие последовательности терминала.



Наш эксперт

Андрей Боровский
Программирует на Pascal с 14 лет, разрабатывает ПО для Linux с 2001 года. Всегда готов поделиться своим богатым опытом с читателями LXF.

Едва ли в наших программистских (и администраторских) рядах сыщется человек, незнакомый с командой **su**. Вопреки распространенному заблуждению, эта аббревиатура означает не «super user» (что в английском языке вообще одно слово), а «switch user», хотя используется данная утилита, в основном, как раз для временного получения привилегий root. Но сегодня нас будет интересовать не этимология, а ее внутреннее устройство: выполнение какой-либо операции «руками суперпользователя» – столь распространенная задача, что не грех научиться решать ее более-менее стандартным образом.

Помимо безопасности и надежности, наши приложения (пусть даже консольные) должны быть по возможности красивыми. В завершение этого урока мы еще раз коснемся вывода на экран цветного текста, на сей раз – не используя ничего, кроме чистого C.

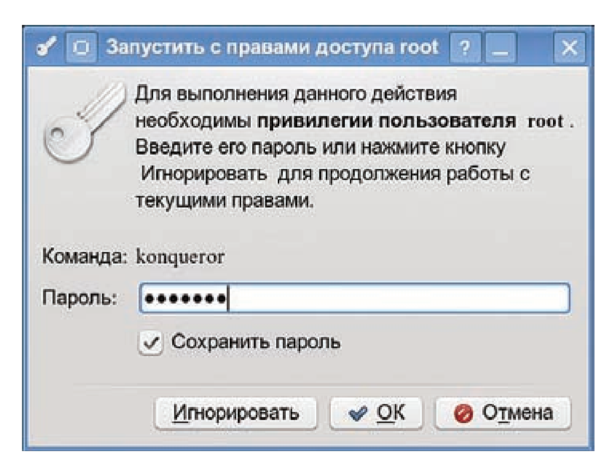
Как root

Все вы, конечно, имели дело с программами, которые позволяют получить доступ к некоторым возможностям root, не покидая учетной записи обычного пользователя: взять ту же *su*. Примерами таких приложений могут также служить утилиты для настройки оборудования и установки программного обеспечения. Внешне все выглядит так, как будто пользователь запускает обычную программу, вводит пароль root и становится (внутри этой программы) суперпользователем. На самом деле, все обстоит немного иначе.

Владельцем любого процесса (экземпляра программы) по умолчанию является запустивший его пользователь. Права процесса на доступ к файлам, привилегированным функциям API и другим объектам системы определяются правами владельца процесса. Соответственно, любая программа, запущенная пользователем root, получает максимальные полномочия. Разработчики Unix довольно быстро почувствовали, что переходить из одной учетной записи в другую каждый раз, когда пользователю требуется выполнить особые действия, неудобно, и придумали расширенные флаги прав доступа setuid и setgid. Если эти биты установлены, владельцем процесса считается владелец файла программы (а не пользователь, который запустил процесс). Иначе говоря, если некий исполняемый файл принадлежит пользователю root и для него установлен флаг setuid, то независимо от того, какой пользователь создаст из этого исполняемого файла процесс, он получит такие же права, как если бы его запустил root. При этом система не забывает, кто запустил процесс на самом деле. Для всех процессов система хранит два набора идентификаторов пользователя и группы – действительные (effective) и фактические (real). Если пользователь vpupkin запустит программу с установленным битом setuid, причем владельцем файла программы является root, действительным хозяином процесса окажется root, а фактическим – vpupkin. В своем взаимодействии с системой программа будет использовать права действительного владельца.

Проблема, которую нам предстоит решить, заключается в том, что мы хотим предоставлять право запуска программ с установленным флагом setuid не каждому пользователю, а только тому, кто знает пароль root. Возникает вопрос: зачем такому пользователю вообще нужны setuid-программы, ведь он может просто зайти в систему как root? На самом деле, возможность запускать setuid-программу и удобна (не нужно специально каждый раз регистрироваться в системе), и безопасна – пользователь проводит в режиме root минимум времени.

Таким образом, классическая схема авторизации пользователя в setuid-программе выглядит так: приложение, запущенное неким пользователем с действительными правами root, запрашивает у пользователя пароль root. Если введен правильный пароль, программа предоставляет пользователю возможность выполнить требуемые действия, в противном случае ему предлагается повторить попытку ввода пароля. Как видим, программы, запрашивающие у пользователя пароль root, не становятся программами суперпользователя в результате ввода пароля – они являются таковыми с самого начала. Поскольку пользователь, запустивший программу с установленным флагом setuid root, уже фактически имеет те права, которые он хочет получить, программы, предназначенные для подобного применения, должны быть написаны максимально аккуратно, чтобы запускающий их человек не мог обойти механизм аутентификации и получить привилегии root, не зная пароля (*см. врезку*). Отметим также, что новейшие средства аутентификации пользователей (PolicyKit) физически разделяют код, выполняющий аутентификацию, и код, выполняющий привилегированные действия. Впрочем, этот вопрос выходит за рамки данной статьи.



› *Kdesu* — еще один способ выполнить какую-либо задачу от имени root, знакомый всем пользователям KDE.

» **Месяц назад** Мы избавлялись от ошибок в программах, используя GNU Debugger.

для паролей



Время кодировать

За время существования Unix и Linux способов аутентификации пользователя было придумано немало, но в основе всего по-прежнему лежит старый добрый файл зашифрованных паролей. Именно с работы с ним мы и начнем.

Ниже приводится листинг программы *goroot*. Она запрашивает у пользователя, запустившего ее, пароль root, и если он правильный, создает файл **rootfile**, владельцем которого будет root (а не пользователь, запустивший программу). Потом программа временно отказывается от прав root и создает файл **userfile**, владельцем которого будет пользователь, запустивший программу (во всем этом можно убедиться с помощью команды **ls -al**). Затем программа возвращает себе полномочия root и создает файл **rootfile2**, владельцем которого снова оказывается root.

```
#define _XOPEN_SOURCE

#include <stdlib.h>
#include <stdio.h>
#include <signal.h>
#include <termios.h>
#include <unistd.h>
#include <sys/types.h>
#include <shadow.h>
#include <sys/types.h>
#include <pwd.h>
#include <sys/stat.h>
#include <fcntl.h>

void read_pwd (char * password, int buf_size)
{
  char ch;
  int i;
  if (!isatty(fileno(stdin))) {
    password[0] = 0;
    return;
  }
  struct termios oldsettings, newsettings;
  tcgetattr(fileno(stdin), &oldsettings);
  newsettings = oldsettings;
  newsettings.c_lflag &= ~(ECHO|ICANON|ISIG);
  newsettings.c_cc[VMIN] = 0;
  newsettings.c_cc[VTIME] = 0;
  tcsetattr(fileno(stdin), TCSANOW, &newsettings);
  i = 0;
  printf(“Введите пароль\n”);
  while((ch = getchar()) != '\n') {
    if ((i<buf_size-1) && (ch != EOF)) {
      password[i] = ch;
      i++;
    }
  }
  password[i] = 0;
  tcsetattr(fileno(stdin), TCSANOW, &oldsettings);
}

int auth_root(char * password)
{
  char * epasswd;
  struct spwd *spwd;
  spwd = getspnam(“root”);
  epasswd = crypt(password, spwd->sp_pwdp);
  return !strcmp(epasswd, spwd->sp_pwdp);
}

#define BUF_SIZE 16

int main(void)
{
  char password[BUF_SIZE];
  int result = 0;
  if (geteuid() != getuid()) {
   read_pwd(password, BUF_SIZE);
   result = auth_root(password);
  }
   else result = 1;
  if (result) {
   int rootuid = geteuid();
   printf(“Теперь Вы root\n”);
   open(“rootfile”, O_CREAT|O_EXCL);
   seteuid(getuid());
   setegid(getgid());
   printf(“А теперь - нет\n”);
   open(“userfile”, O_CREAT|O_EXCL);
```

»

Пароль под контролем

Функция read_pwd() играет в нашей программе служебную роль; тем не менее, она гораздо длиннее, чем auth_root(). Функция, предлагающая ввести пароль, зачастую становится первым объектом атаки злоумышленников. Если взломщику удастся обмануть ее, он получит доступ к программе, не зная пароля. Хуже того, в некоторых случаях взломщик, сломавший функцию ввода пароля, может получить даже более широкие права, чем те, которые предоставляет данная программа по умолчанию: например, командную строку суперпользователя. По этой причине очень важно, чтобы suid-программы максимально ограничивали свободу действий даже тех пользователей, которые прошли процедуру аутентификации. Разумеется, ограничивать свободу действий пользователя, который действительно знает пароль root, бессмысленно, но никогда не следует исключать возможность проникновения «пользователя с черного хода».

Функция read_pwd() делает все возможное для того, чтобы пользователь не мог ее обойти. Сколько бы символов ни ввели с клавиатуры, буфер password будет заполнен не более чем buf_size символами. Вводимые символы, естественно, не отображаются на экране.

› Наше первое приложение. Смотреть не на что, но так и задумано: кто же выводит пароль root на экран?

```
        seteuid(rootuid);
        printf("А теперь - снова root\n");
        open("rootfile2", O_CREATIO_EXCL);
    } else {
        printf("Неверный пароль\n");
    }
    return EXIT_SUCCESS;
}
```

Давайте начнем разбор программы с функции main(). Прежде всего надо убедиться, что пользователь, запустивший программу – не root (действительно, зачем спрашивать пароль root у самого root?). Мы проверяем этот факт очень просто – сравнением значений функций getuid() и geteuid(). Первая функция возвращает фактический идентификатор поль зователя, вторая – действительный. Поскольку у программы с установленным setuid root действительный идентификатор всегда соответствует идентификатору root, его совпадение с фактическим означает, что программу запустил root.

«Интереснее всего бывает, если программу запускает не root.»

Интереснее всего, конечно, бывает, если пользователь, запустивший программу – не root. В этом случае мы предлагаем ему ввести пароль (функция read_pwd()) и проверяем корректность предоставленных им сведений в функции auth_root().

Аутентификатор-1

Чтобы понять, как сравнивать введенный пароль с паролем пользователя root, следует знать, каким образом последний хранится в Unix-системах. Пароль каждого пользователя записан на диске в зашифрованном виде, причем алгоритм, используемый для шифрования, является однонаправленным, то есть не предусматривает расшифровки. Чтобы сравнить введенный пароль с хранимым, его нужно зашифровать тем же способом, что и хранимый пароль, а затем сравнить результаты. Таким образом, даже получив файл паролей, злоумышленник не сможет воспользоваться им напрямую: ведь у него будут одни только шифры. При вводе такого «пароля» в программу он будет зашифрован еще раз, и результат, естественно, не совпадет с ожидаемым. Первоначально эта схема считалась настолько безопасной, что файл, хранящий зашифрованные пароли пользователей, был доступен для чтения всем желающим. Вскоре, однако, выяснилось, что, имея на руках файл с паролями и достаточно мощный компьютер, можно довольно быстро найти пароли методом целенаправленного подбора. В результате были приняты некоторые дополнительные меры предосторожности, вдаваться в подробности которых мы здесь не будем. Для нас алгоритм аутентификации пользователя выглядит так: получить зашифрованный пароль пользователя root; зашифровать пароль, введенный пользователем программы; сравнить результаты.

Зашифрованный пароль любого пользователя системы можно получить с помощью функции getspnam(). Ее аргументом является имя пользователя, а возвращаемым значением – указатель на структуру spwd, которая, помимо прочего, содержит зашифрованный пароль пользователя (поле sp_pwdp). Следует иметь в виду, что сама функция getspnam() сработает только в том случае, если владельцем вызывающего ее процесса является привилегированный пользователь – root или член группы shadow. Если владелец процесса не удовлетворяет этим требованиям, функция возвращает значение NULL. Таким образом, наша программа *goroot* должна обладать действительными правами root, чтобы как минимум получить доступ к базе паролей.

Шифрование введенного пользователем пароля выполняется функцией crypt(). Она принимает два аргумента: строку пароля с нулевым окончанием и «затравку» – строку, которая указывает параметры алгоритма, используемого для шифрования пароля. Раньше вторым аргументом функции crypt() были байты (крупицы?) «соли» (salt) – случайные символы, которые добавлялись к выбранному пользователем паролю и затрудняли «атаку со словарем». Теперь вторым аргументом crypt() может быть любая информация о том, как зашифрован пароль. Откуда нам взять эту информацию, если учесть, что введенный пользователем пароль должен быть зашифрован точно так же, как и хранящийся в системе пароль root? Оказывается, первые байты строки spwd->sp_pwdp как раз и содержат требуемые сведения, так что в качестве второго аргумента crypt() мы используем строку spwd->sp_pwdp. Зашифровав полученный от пользователя пароль, остается только сравнить его с сохраненным паролем root с помощью функции strcmp().

Если пользователь прошел аутентификацию, мы поздравляем его с этим фактом, а затем создаем файл **rootfile**. Поскольку действительным владельцем процесса является root, владельцем созданного файла будет тоже root. Допустим, что теперь мы хотим, чтобы программа временно отказалась от полномочий root. Самый простой способ сделать это – установить действительные идентификаторы пользователя и группы равными фактическим идентификаторам, которые соответствуют идентификаторам пользователя, запустившего процесс. Именно это мы и делаем:

```
seteuid(getuid());
setegid(getgid());
```

Теперь действительным владельцем процесса числится запустивший его пользователь, и привилегии процесса соответствуют привилегиям этого пользователя. С помощью функции seteuid() мы можем снова сделать пользователя root эффективным владельцем процесса, когда нам это понадобится.

Программу goroot следует компилировать с ключом -lcrypt:

```
gcc goroot.c -o goroot -lcrypt
```

Помня, что программа будет работать, только если ее владелец – root, да еще и установлен флаг suid, в режиме root скомандуем:

```
chown root goroot
chmod ug+s goroot
```

Теперь наше первое приложение готово к работе.

Аутентификатор-2

Пароль в наше время – не единственное средство аутентификации пользователя. Соответствие между именем учетной записи и реальным человеком можно установить многими способами, в том числе путем сканирования отпечатков пальцев или радужной оболочки глаза (некоторые уверяют, что можно использовать вырванное око своего врага, но я все же думаю, что врага придется тащить целиком). Учитывая многообразие средств аутентификации, в современных системах этот процесс абстрагирован от конкретной программы, для чего применяются подключаемые модули аутентификации (Pluggable Authentication Modules, PAM). Вот как выглядит аналог программы *goroot*, написанный с применением PAM:

```
#include <security/pam_appl.h>
#include <security/pam_misc.h>
#include <stdio.h>

int main ()
{
  pam_handle_t* pamh;
  struct pam_conv pamc;
  pamc.conv = &misc_conv;
```

```
  pamc.appdata_ptr = NULL;
  pam_start (“common-auth”, “root”, &pamc, &pamh);
  if (pam_authenticate (pamh, 0) != PAM_SUCCESS) {
   printf(“Неверный пароль\n”);
   pam_end (pamh, 0);
   return 1;
  }
  printf (“Теперь Вы - root.\n”);
  open(“rootfile”, O_CREATIO_EXCL);
  pam_end (pamh, 0);
  return 0;
}
```

Программа взаимодействует с подсистемой PAM в режиме транзакций. Транзакция начинается с помощью функции pam_start(), первый аргумент которой – это имя сервиса PAM. Для серьезной программы в процессе настройки системы будет создан свой сервис (описания сервисов обычно «живут» в директории **/etc/pam.d**). Мы используем сервис *common-auth*, который является базовым, предназначенным для аутентификации пользователей. Второй аргумент функции pam_start() – имя пользователя, в нашем случае – root. Далее следуют указатели на две структуры, описывающие состояние процесса аутентификации. Сама аутентификация выполняется с помощью функции pam_authenticate(), которая использует метод аутентификации, принятый в вашей системе (скорее всего, это будет все то же приглашение для ввода пароля). По крайней мере, теперь мы можем переложить ответственность за создание функции чтения пароля на разработчиков системы.

В случае, если аутентификация прошла успешно, функция pam_authenticate() возвращает значение PAM_SUCCESS, с чем мы пользователя и поздравляем. Компилировать программу следует с ключами **-lpam** и **-lpam_misc**. После сборки исполняемый файл программы должен быть передан во владение root, и для него должен быть установлен флаг setuid, как и в предыдущем случае. Между прочим, программе, использующей PAM, по-прежнему необходим доступ к файлу паролей, а значит, она не сможет выполнить аутентификацию пользователя, если не будет запущена с действительными правами root.

Да будет цвет!

Все мы видели консольные программы, в которых различные важные надписи выделяются цветом. Один из способов раскрасить экран – использовать библиотеку *ncurses*. Однако использование *ncurses* только для вывода разноцветных надписей – это, что называется, из пушки по воробьям. Помимо прочего, библиотека заменяет стандартные параметры консоли своими собственными, и это может быть неприемлемо для нас. Между тем, у нас есть простой способ выводить на экран выделенные цветом надписи (а также проделывать многие другие манипуляции с экраном), не прибегая к помощи *ncurses*. С тех пор как терминалы обзавелись расширенными возможностями, был придуман способ управления этими возможностями, которым могла воспользоваться любая программа, выводящая данные на терминал: специальным командам терминала соответствовали специальные последовательности символов – Esc-последовательности. Рассмотрим пример программы:

```
#include <stdio.h>

int main() {
   int i;
   printf(“\033[30mЧерный \033[0m\n”);
   printf(“\033[31mКрасный \033[0m\n”);
   printf(“\033[32mЗеленый \033[0m\n”);
   printf(“\033[33mКоричневый \033[0m\n”);
   printf(“\033[34mСиний \033[0m\n”);
   printf(“\033[35mФиолетовый \033[0m\n”);
   printf(“\033[36mГолубой \033[0m\n”);
   printf(“\033[37mСерый \033[0m\n”);
   printf(“\033[37;1mБелый \033[0m\n”);
   printf(“\033[33;1mЖелтый \033[0m\n”);
  return 0;
}
```

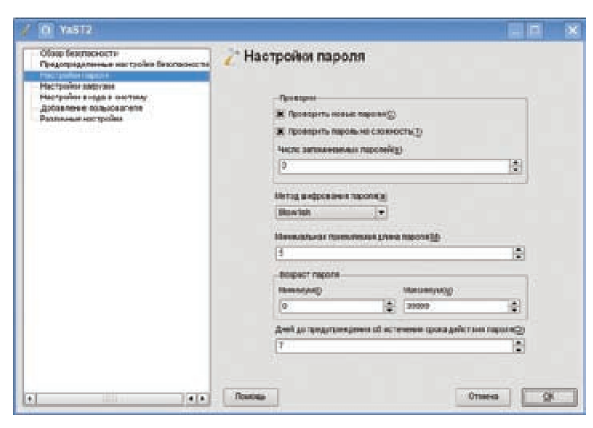
Эта программа распечатывает на экране терминала надписи, цвет каждой из которых соответствует напечатанному слову (на самом деле, цвет, который вы увидите на экране, будет соответствовать выбранной вами цветовой схеме). Командная последовательность начинается с префикса **\033**[(**\033** – код символа Esc). Формат команды для смены цветов выглядит так:

```
Esc[код_цвета<;дополнительный_атрибут>m
```

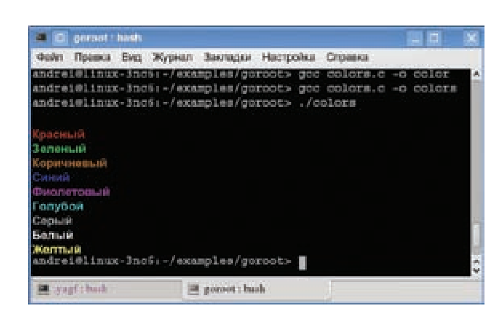
Команда заканчивается символом **m**. Кодам 8-ми базовых цветов соответствуют числа 30–37. Команда со специальным кодом цвета 0 возвращает терминал в стандартное состояние. Дополнительный (необязательный) атрибут позволяет установить такие параметры, как яркость или мерцание (с помощью бита яркости из 8 базовых цветов можно получить 7 дополнительных). Esc-последовательности позволяют не только изменять цвет символов и фона. Вот как, например, выглядит функция, перемещающая курсор в позицию, заданную координатами x (столбец) и y (строка):

```
void move(int x, int y) {
   printf(“\033[%i;%iH”, y, x);
}
```

Ладно, скажете вы, но где гарантия, что эти командные последовательности будут работать так же и в других системах? Когда создавалась библиотека *ncurses*, такой гарантии действительно не было, но теперь стандартизация сделала нашу жизнь проще. Приведенные выше последовательности стандартизированы в рамках спецификации ECMA-48 (можете ознакомиться с полным текстом стандарта, если хотите узнать об управляющих последовательностях терминала побольше). Стандарт ECMA-48 поддерживают все эмуляторы терминалов Linux и других Unix-систем. **LXF**



› **Настройка параметров пароля в OpenSUSE: алгоритмы шифрования можно выбрать самые разные.**



› **Разноцветный текст без *ncurses* — это возможно.**

» Через месяц Ознакомимся с Lua и разберемся, за что его уважают разработчики игр.

Безопасность: Для

Мартин Мередит научит вас управлять портами, разбираться с уязвимостями и отбивать хакеров, норовящих пролезть к вам в систему.





Наш эксперт

Мартин Мередит – разработчик Debian и Ubuntu, а также эксперт по безопасности для основных торговых онлайн-сетей Великобритании.

В былые времена телефонных модемов Интернет в основном содержал библиотечные каталоги, военные секреты и студенческие страницы с похождениями для Dungeons & Dragons. Ныне существуют сайты для людей, их домашних питомцев, друзей, семьи и бизнеса. Большинство людей охотно пользуются бесплатным хостингом или платят компаниям за размещение своих сайтов, но подлинные web-мастера предпочитают выделенный или виртуальный частный сервер [Virtual Private Server, VPS].

Запуск собственного сервера подразумевает осведомленность о множестве потенциальных проблем с безопасностью, которым вы подвержены в Интернете. В наши дни на большинстве домашних компьютеров установлен брандмауэр, или соединение идет через маршрутизатор, способный защитить от таящихся в Сети угроз. Если у вас собственный сервер, вам все равно нужен брандмауэр, но ради безопасности можно сделать гораздо больше, и мы покажем вам, как.

Защищены ли вы?

Каждая служба, к которой вы подключаетесь в Интернете, пользуется своим портом. Например, соединяясь с web-сайтом, вы используете порт 80 (или порт 443 для HTTPS), а подключаясь по SSH – порт 22. FTP использует порт 21, IMAP – порт 143, и так далее. Когда сервер запущен, он открывает эти порты и ждет входящих соединений.

Но как это отражается на вас? Владея сервером, вы можете иметь несколько разных работающих служб – может быть, у вас будут службы базового стека LAMP или электронной почты. Эти службы, как правило, открывают свои порты для всех пользователей Интернета, что не всегда хорошо.

В качестве примера исследуем свежеразвернутый сервер с помощью программы *nmap*. Она обычно имеется в менеджере пакетов, или в крайнем случае по адресу http://nmap.org. Установив *nmap*, запустите ее с адресом вашего сервера, чтобы получить примерно такой вывод:

```
mez@lazy: % nmap torpor
Starting Nmap 4.76 ( http://nmap.org ) at 2009-05-04 11:56
BST
Interesting ports on torpor:
Not shown: 984 closed ports
PORT     STATE   SERVICE
22/tcp   open    ssh
25/tcp   open    smtp
53/tcp   open    domain
80/tcp   open    http
110/tcp  open    pop3
143/tcp  open    imap
993/tcp  open    imaps
1234/tcp open    hotline
3306/tcp open    mysql
10000/tcp open   snet-sensor-mgmt
Nmap done: 1 IP address (1 host up) scanned in 3.46
seconds
```

Результат: *nmap* выявил 10 портов, доступных в Интернете всем подряд. Большинство из них и должны быть открыты, если сервер, например, предоставляет SSH или почту. Однако, скажем, к *MySQL* доступ из Сети совершенно лишний, а некоторые порты (в частности, 1234) вообще открыты непонятно кем и зачем.

Меня кто-то слушает?

Итак, как мы увидели выше, не всегда можно сказать, кто прослушивает определенный порт. Быстрый поиск в Google разъяснит нам, что порт 1234 облюбовали всякие паразиты типа троянов, а значит, придется провести расследование и выяснить, что происходит и почему данный порт стоит открытым.

Простейший способ узнать, какие программы слушают порты – вызов команды **netstat**. Запуск ее без параметров предоставит список открытых в данный момент соединений. Но нам-то хотелось бы узнать, какая конкретная программа прослушивает конкретный порт (наш 1234).

Для ее поиска запустим **netstat -pnl** от имени root. Выдастся на первый взгляд бессмысленный список, но нас интересуют в нем только две колонки. Номер порта нам известен; найдем колонку с именем Local Address, а в ней – запись, оканчивающуюся

» **Месяц назад** Мы создали и настроили домашнюю АТС на базе *Asterisk*.

вашего сервера

на этот номер, :**1234** (двоеточие разделяет IP-адрес и номер порта). Подходящая строка выглядит так:

```
tcp    0    0 0.0.0.0:1234    0.0.0.0:*    LISTEN
24481/php-cgi
```

В последней колонке – искомая программа. Итак, *php-cgi* слушает все IP-адреса на сервере (0.0.0.0 означает все IP) для любых подключений. PHP – это скриптовый язык, и здесь он установлен в режиме Fast CGI для прослушивания порта 1234.

Разрешать кому попало запускать скрипты на нашем сервере – идея не очень хорошая, так что пора заручиться помощью *iptables*.

Насаждающий законы

Доступный по умолчанию во всех современных дистрибутивах, *iptables* – это ответ Linux на все ваши запросы к брандмауэру (или межсетевому экрану). Он располагается между компьютером и Интернетом, разрешая или запрещая трафик, поступающий в ваш сервер или из него, на основе набора правил *iptables*.

Iptables относительно прост в настройке, но изучение написания правил требует времени. Ниже приведен файл с правилами, которые мы применим на нашем сервере.

```
*filter
:INPUT DROP [0:0]
:FORWARD DROP [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state RELATED,ESTABLISHED –j ACCEPT
-A INPUT -s 127.0.0.1 -d 127.0.0.1 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 25 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 53 -j ACCEPT
-A INPUT -p udp -m udp --dport 53 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 110 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 143 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 993 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 10000 -j ACCEPT
-A INPUT -s 192.168.1.3 -p tcp -m tcp --dport 3306 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 0 -j ACCEPT
-A INPUT -p icmp -m icmp --icmp-type 8 -j ACCEPT
COMMIT
```

***filter** говорит нам, к какой таблице обращаются эти правила. Мы здесь займемся только таблицей фильтрации, но есть и другие таблицы, для настройки, например, NAT (Network Address Translation), маршрутизации и так далее.

Следующие несколько строк, содержащие **[0:0]**, устанавливают политики по умолчанию для цепочек в *iptables*. Используя *iptables*, мы в основном работаем с тремя цепочками: INPUT для всех входящих соединений, FORWARD для всех соединений, пересылаемых на другие сервера (ими мы заниматься не будем), и OUTPUT для всех исходящих соединений. Политики для этих цепочек могут быть установлены как ACCEPT, DROP или REJECT, что означает разрешить соединение, игнорировать соединение или отослать назад код ошибки, говорящий, что порт не открыт. Здесь мы по умолчанию установим цепочки INPUT и FORWARD в DROP для всех соединений, а цепочку OUTPUT – в ACCEPT. Однако умолчания применяются только тогда, когда пакет не соответствует ни одному из правил, и следует задать несколько исключений.

Прежде чем это сделать, упомянем о финальной строке, **COMMIT**: она велит *iptables* применить правила к брандмауэру. В этом пункте ваш межсетевой экран активируется и встанет на стражу сервера.



Запуск **netstat -pnl** от имени root сообщит вам, какие программы слушают порты.

Заткнем дыры

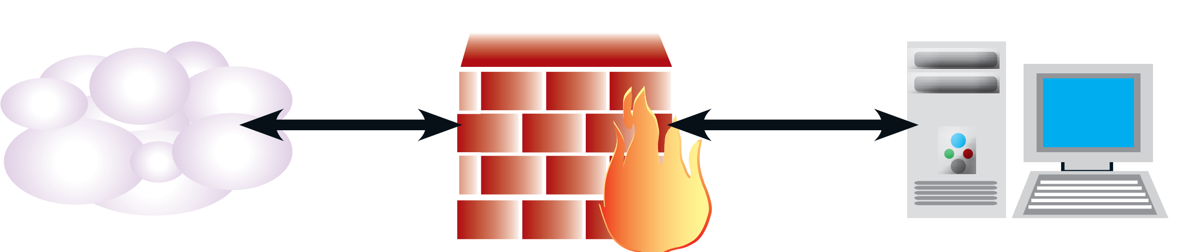
Строки, начинающиеся на **-A INPUT**, устанавливают правила брандмауэра. Здесь они записаны так, как если бы вы вызывали *iptables* напрямую, но без имени *iptables* в начале команды. Подробную информацию о том, как писать правила, ищите в man-странице *iptables*.

Каждое правило начинается с **-A INPUT**, чтобы *iptables* применил его к цепочке входящих пакетов. Если мы хотим применить правило к исходящим соединениям, вместо **INPUT** надо поставить **OUTPUT**. После этого идет описание пакетов, подпадающих под правило, и заканчивается строка на **-j ACCEPT**, чтобы *iptables* присоединил пакет к цепочке **ACCEPT** – другими словами, пропустил его. Если мы хотим применить для пакета **REJECT** или **DROP**, то вместо **ACCEPT** так здесь и запишем.

Теперь рассмотрим строку

```
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

которая велит *iptables* разрешить любые соединения, которые »



› Вы видите эту диаграмму раньше, чем я про нее рассказал — все и вправду так просто.

установлены (ESTABLISHED) или связаны (RELATED) с установленными. Это хорошее правило, поскольку оно разрешает продолжить любые уже созданные соединения – к примеру, существующее SSH-соединение будет оставаться активным, если вы используете SSH для изменения правил брандмауэра. Оно разрешает также входящие соединения, имеющие отношение к исходящим соединениям (TCP работает в обоих направлениях, и без этого правила может оказаться, что вам недоступны исходящие соединения с компьютера).

Следующая за ней строка разрешает *iptables* принимать любые соединения, выполняемые с 127.0.0.1 (**-s** означает IP-адрес источника) на 127.0.0.1 (**-d** ссылается на IP-адрес получателя); это можно перевести как «принимать любые соединения от самого себя».

В следующих девяти строках мы открываем порты, нужные нам для связи с внешним миром. Заметьте, что у нас нет разрешения доступа к порту 3306 (*MySQL*) и 1234 (ошибочная конфигурация PHP): оно учтено в правиле брандмауэра, позволяющем локальные соединения. А вот что не мешает добавить – это сброс всех внешних попыток доступа к *MySQL*, такого вида:

```
-A INPUT -s 192.168.1.3 -p tcp -m tcp --dport 3306 -j ACCEPT
```

Здесь IP-адресу 192.168.1.3 разрешено присоединяться к порту 3306, потому что в нашей локальной сети есть еще один сервер, нуждающийся в доступе к *MySQL*.

Две строки прямо над **COMMIT** позволяют серверу распознавать пинги и включают отправку ответов на них.

После сохранения файла (куда нибудь типа **/etc/iptables.conf**) можно запустить команду

```
iptables-restore < /etc/iptables.conf
```

включив правила брандмауэра, написанные ранее. Повторно запустив *nmap*, мы получим следующий результат:

```
mez@lazy: % sudo nmap torpor
Starting Nmap 4.76 ( http://nmap.org ) at 2009-05-04 11:56
BST
Interesting ports on torpor:
Not shown: 984 filtered ports
PORT  STATE  SERVICE
22/tcp  open  ssh
25/tcp  open  smtp
53/tcp  open  domain
80/tcp  open  http
110/tcp  open  pop3
143/tcp  open  imap
993/tcp  open  imaps
10000/tcp  open  snet-sensor-mgmt
Nmap done: 1 IP address (1 host up) scanned in 10.30
seconds
```

Заметим, что вывод теперь сообщает, что отфильтрованные порты не показываются взамен закрытых (строка 4). Также может оказаться, что *nmap* надо запустить от имени root – в нашем тесте эта конфигурация брандмауэра работала чересчур бдительно, и *nmap* не видел все открытые порты, пока его не запускали от имени root.

Скорая помощь

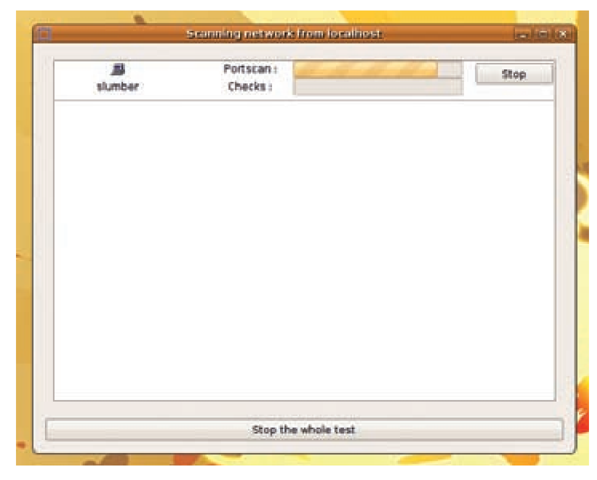
Вы можете запускать правила брандмауэра при старте системы, добавив команду **iptables- restore** в **/etc/rc.local**.

Скажи «Пароль»!

Даже с применением многочисленных мер безопасности все еще сохраняется шанс, что кто-то взломает ваш сервер, просто угадав пароль! Существуют скрипты, которые, применяя словарь, могут перебрать все возможные комбинации имени пользователя и пароля за относительно короткое время.

Fail2ban (www.fail2ban.org) пробует предотвратить это, отслеживая наличие сбоев входа в файле журнала. Обнаружив, что кто-то несколько раз подряд безуспешно пытается войти, он на некоторое время запрещает такому старателю доступ на сервер (используя брандмауэр).

Это, конечно, не слишком понравится тем, кто регулярно забывает пароль, зато остановит попытки лобовых словарных атак на ваш сервер.



› *Nessus* сканирует ваш сервер на уязвимости в его коде, которыми могут воспользоваться недобросовестные лица.

Что такое Nessus?

Простой брандмауэр должен отчасти защитить нас, но в любой системе существуют и другие потенциальные уязвимости, особенно в серверах, поскольку они обновляются не часто. Для борьбы с этим создан инструмент, известный под именем *Nessus*. Он состоит из двух частей: клиента и сервера, по той причине, что сервер может быть установлен удаленно и использоваться для проверки соединения с локальной службой. В нашем случае компьютер, который мы подвергаем тестированию, не является для нас локальным, поэтому мы можем установить сервер и клиент локально.

Установив *Nessus* и *nessud* с помощью вашего менеджера пакетов, создайте пользователя. Запустите команду **nessus-adduser** и следуйте инструкциям на экране. Теперь запустите клиент *Nessus* (его можно найти в Ubuntu в меню Приложения > Интернет > Nessus).

Запущенный *Nessus* работает со списком модулей расширения, которые предназначены для проверки уязвимостей. Так как новые уязвимости находят каждый день, *Nessus* необходимо постоянно обновлять. Tenable Network Security, создатель *Nessus*, имеет два разных списка модулей – HomeFeed и ProfessionalFeed. ProfessionalFeed предоставляет более свежий список, включая самые последние уязвимости, которые можно протестировать сразу же после появления соответствующего модуля. Правда, ProfessionalFeed стоит 1200 долларов США, и если вы работаете с *Nessus* не в режиме ежедневных проверок и не в сверхсекретном учреждении, HomeFeed будет достаточно.

Для скачивания своих модулей Tenable требует регистрации, которую можно произвести на http://linkpot.net/enviably. После этого запустите команду **nessus-fetch --register <ваш регистрационный код>**, где нужно вставить код, полученный на вашу электронную почту после регистрации.

Настроив и загрузив программу, вы можете обнаружить, что основное окно *Nessus* немного запутано. Чтобы вы не блуждали в нем бездумно, мы покажем, как выполнить простое сканирование.

Первым делом зайдите на сервер с данными, которые вы задали ранее. После подключения перед вами предстанет окно Plugins. Это список скриптов, которые *Nessus* попытается запустить. Нажмите на кнопку Select All и переключитесь на вкладку Scan Options. Здесь находятся основные параметры сканирования.

Самая интересная из них – флажок Safe Checks. При его установке *Nessus* не будет выполнять сканирование, потенциально способное обрушить ваш сервер. Однако если у вас есть физи-

ческий доступ к серверу (или какой-либо способ перезагрузить его удаленно), рекомендуем снять этот флажок, тогда проверка будет более углубленной. В конечном счете лучше заставить *Nessus* найти дыру в безопасности и обрушить сервер, чем оставить зияющую уязвимость.

Наконец, задайте имя или IP-адрес сервера на вкладке Target, а затем уступите место *Nessus* простым нажатием Start The Scan и ждите, пока программа сделает все остальное.

Завершив сканирование, *Nessus* представит вам отчет с указанием перечня найденных проблем. Многие сообщения *Nessus* относятся к информационным, типа «у вас есть работающий SSH-сервер», но сообщения особой важности помечаются красной иконкой STOP рядом с ними. Найдя уязвимости, *Nessus* подскажет вам, как их исправить (или направит в такое место, где можно узнать решение). В случае критических проблем вы должны действовать немедленно.

Поставить растяжку

Если все пойдет хорошо, то после сканирования *Nessus* и последующего закрытия уязвимости на ваш сервер будет трудно попасть. Однако это не означает, что нельзя нарушить вашу безопасность: единственный способ добиться, чтобы ни один злоумышленник не мог войти в вашу систему – это не включать сервер вообще.

Итак, что же делать, если гарантией безопасности является только отключение системы – или, по крайней мере, отключение от Интернета? На тот момент, когда хакер уже влез в вашу систему, вы все еще можете, как минимум, убедиться в его присутствии. Это позволит сделать система обнаружения вторжений (IDS), хотя мы надеемся, что она вам никогда не пригодится.

Сейчас самая известная IDS – это *Tripwire*, которая разрабатывается с 1992 года. В данный момент существуют три программы с таким именем, и все они делают одно и то же. И только одна из них относится к открытому коду. Вы можете найти ее в вашем дистрибутиве или на сайте http://tripwire.sf.net.

Tripwire работает, создавая базу данных всех файлов вашей системы и отмечая, когда в них происходят изменения. Из-за особенностей работы системы, лучше всего установить *Tripwire* до подключения компьютера к Интернету. Если кто-то уже дорвался до вашей машины раньше, чем программа была установлена, то *Tripwire* в итоге просто даст уверенность, что черные ходы, созданные злоумышленником, остаются в силе.

> «Лучше установить Tripwire до подключения к Интернету.»

Установка *Tripwire* – процесс длительный и выходит за рамки данного урока, но вы можете найти более подробную информацию о процессе на www.alwanza.com/howto/linux/tripwire.html и www.tripwire.com.

Приглядим за серверами

Возможно, самое важное, что нужно сделать при рассмотрении безопасности вашего сервера, это убедиться, что вы видите все, что на нем происходит. Этот процесс также известен как аудит, и, правильно организованный, он может гарантировать, что проблемы будут решены прежде, чем они приобретут катастрофические пропорции.

Мы уже использовали *Nessus* для аудита безопасности снаружи, и упомянули о проверке целостности файлов в системе с помощью *Tripwire*, но есть много и других инструментов аудита. Однако мы находим, что лучший способ аудита системы – чтение файлов журналов; или уж заполучите программу, которая сделает это за вас (подробнее об этом ниже).

Серверы склонны генерировать большой объем информации о происходящем. Посмотрите в директорию **/var/log**: вы найдете массу журнальных файлов, от системного журнала до журнала доступа *Apache*. Журналы предоставляют много полезной информации, но определить, что стоит читать, а что нет, не так-то просто.

И здесь на помощь приходит *Logwatch* – эта утилита читает ваши файлы журналов и может отправлять вам ежедневные сообщения о наиболее интересных их частях. Сообщения бывают длинноваты, но они дадут вам знать, если что-то пойдет не по плану. Кроме того, они информируют, когда кто-то из ваших пользователей пытается влезть куда не следует, давая вам время на решение этого вопроса, прежде чем начнутся проблемы.

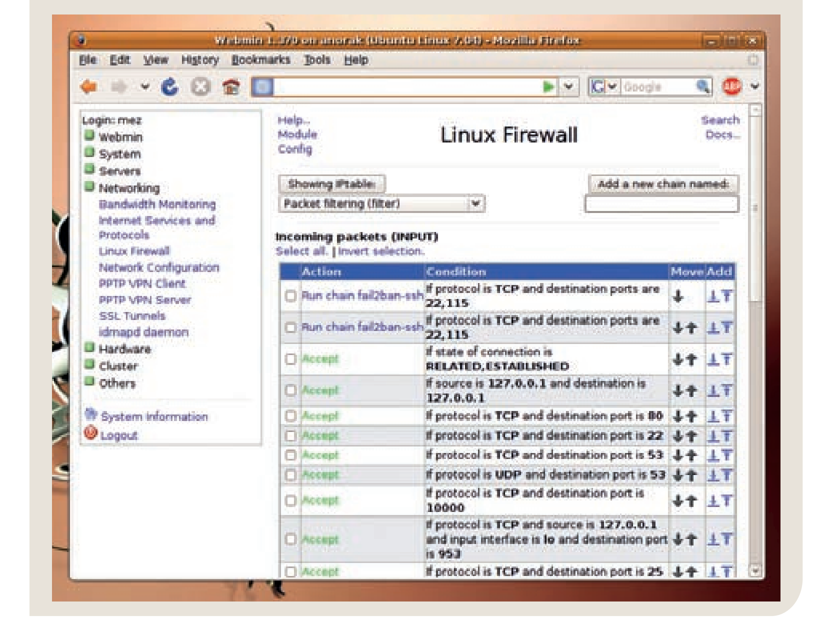
Под Debian и Ubuntu установка и настройка *Logwatch* проста. Однако чтобы принимать информацию в формате HTML, надо задать метод вывода, выходной формат и адрес электронной почты, куда присылать информацию. Вы можете изменить эти параметры, отредактировав **/usr/share/logwatch/default.conf/logwatch.conf**, например, так:

```
Output = mail
Format = html
MailTo = youraddress@yourdomain.net
```

Отчеты будут поступать на ваш ящик, позволяя вам быть в курсе дел на сервере, где бы вы ни были.

Альтернатива командной строке

На данном уроке мы отвели время на изучение написания правил брандмауэра в командной строке. Однако существуют альтернативы такому подходу. Наш фаворит (и мы регулярно его используем) – это *Webmin*, инструмент, позволяющий управлять сервером через web-браузер. Если вы установили *Webmin* на ваш сервер, то найдете опции брандмауэра в Networking > Firewall. Если вы разобрались в рассмотренном здесь создании правил брандмауэра, работа с *Webmin* будет сродни прогулке в парке. Для начала нажмите Revert Configuration – загрузятся ранее созданные правила брандмауэра. Затем правила может будет изменять в зависимости от вашей новой установки *Webmin*. Наконец, покончив с реформами, нажмите Apply Configuration для ввода модифицированных правил в действие.



» **Пропустили номер?** Узнайте на с. 103, как получить его прямо сейчас.



Ответы

Есть вопрос по Open Source? Пишите нам по адресу answers@linuxformat.ru!

В этом месяце мы ответим на вопросы про:

1 Пустой дом

В После обновления Ubuntu до 8.04 с ядром linux-2.6.24-23-generic и Gnome 2.22.3 на моем Dell Inspiron 6400 появились мелкие проблемы. Файловый менеджер Gnome зависает при попытках отображения содержимого моей домашней директории. При просмотре других каталогов в системе все работает отлично.

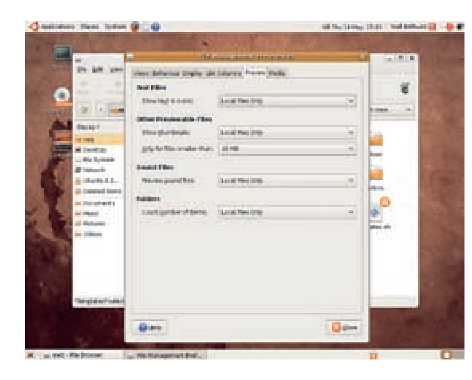
***Audacity* не проигрывает записи. Я перепробовал уже все возможные настройки устройства вывода. Во время обновления до 8.04 система выдавала предупреждение о том, что менеджер пакетов поврежден — что-то насчет невозможности настройки *hplip*.**

Джон Спрэгг [Jon Spragg]

Сколько времени Вы отводите файловому менеджеру на отображение содержимого домашнего каталога? При включенной опции предпросмотра файлов, менеджер может задержаться на создании миниатюр для одного или более из них, особенно если они находятся на разделяемом сетевом ресурсе, а не на локальной машине. Для проверки можно либо совсем отключить предпросмотр в Параметрах файлового менеджера, установив для каждого типа файлов опцию Never [Никогда], либо как минимум убедиться, что эта опция включена только для локальных файлов с небольшим максимально разрешенным размером.

Если это сработает, можно создать временный каталог и переносить в него файлы и подкаталоги из Вашей домашней директории, пока зависания не прекратятся. Тогда Вы узнаете, какой файл или файлы были причиной проблемы, и примете необходимые меры.

Можете ли Вы проиграть файлы в других программах, или сохранить файл в *Audacity*, а затем воспроизвести его другим проигрывателем? Попробуйте это сделать, чтобы определить, таится ли источник проблемы именно в *Audacity* или же воспроизведение звука не работает глобально. Запустите программы, отказывающиеся воспроизвести звук, из консоли: это не заставит их проиграть файл, но зато Вы увидите сообщения об ошибках.



› На создание иконок предпросмотра нужно время; лучше ограничиться небольшими локальными файлами.

Ошибка менеджера пакетов, возможно, произошла из-за поврежденного пакета или пакета с неправильной подписью. Проблему, скорее всего, решит обновление списка пакетов в *Synaptic* и установка некоторых свежих версий пакетов.

Наши эксперты

» Мы найдем ответы на любой вопрос — от проблем с установкой системы или модемом до сетевого администрирования, главное — спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.



Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.



Грэм Моррисон
Когда он не обозревает кучи программного обеспечения и не халтурит с MythTV, Грэм готов дать ответ касательно любого оборудования и проблем виртуализации.



Владимир Житомирский
Председатель Совета директоров Правового бизнес-бюро «Граф Маевский» и убежденный линуксоид; помогает нам в качестве юриста.



Юлия Дронова
Если компьютер у Юлии не занят выполнением команды *emerge*, она спешит применить его для модерирования ЛинуксФорума.

КУДА ПОСЫЛАТЬ ВОПРОСЫ
Пишите нам по адресу: answers@linuxformat.ru **или спрашивайте на форуме:** www.linuxformat.ru

2 Диск Eee... не жилец?!

В Недавно я поставил Ubuntu Netbook Remix 9.04 на свой Eee PC 900. Мне нравится гибкость Ubuntu и доступность обновлений, и я также очень полюбил интерфейс NBR, который на удивление классно работает после незначительных сделанных изменений, включая наложение заплатки на ядро. Но я начитался заметок в Интернете, выражающих беспокойство о том, как NBR (и альтернативные дистрибутивы) работают со встроенным твердотельным диском (SSD). При установке подобных дистрибутивов советуется принимать всяческие меры, как то:

1 Не использовать журналируемую файловую систему на разделах SSD.
2 Не помещать на SSD раздел подкачки.
3 Сразу после установки редактировать *fstab* и указывать noatime **в опциях монтирования SSD.**
4 Не вести журнал системных сообщений или ошибок на SSD.

Также много и других советов по части поведения некоторых программ, например, кэша *Firefox*. В вашем учебнике в номере LXF109 вроде ничего не говорилось о подобных проблемах с SSD при замене ОС на Eee. Не могли бы вы сказать, имеют ли под собой почву какие-нибудь из этих утверждений (или даже все)? Странно, что уста-

новка NBR по умолчанию создает на Eee 900 файловую систему ext3 и раздел подкачки и монтирует разделы с опцией *relatime*. Неужели Canonical не заботится о долговечности SSD? Или все эти страхи надуманы?
Марк Флорис [Marc Floris]

Страхи основываются на том факте, что диски SSD – по существу, флэш-память, срок жизни у которой ограничен. Но используются они не так, как флэш-память на USB-брелках и в картах памяти. Проблемы флэш-памяти связаны с тем, что каждая ячейка выдерживает ограниченное количество циклов перезаписи; для устройств USB и карт памяти это от 100000 до миллиона, в зависимости от качества устройства. Вроде немало. Но отдельные участки диска перезаписываются очень часто: например, таблицы FAT и журналы файловых систем. Поврежденный журнал можно восстановить, но повреждение таблицы разделов на FAT практически означает гибель устройства, а именно эта файловая система используется на переносных USB-устройствах.

Жесткие диски SSD по ряду причин отличаются. Как правило, в них используются более качественные компоненты. На нетбуках не применяется файловая система FAT. А главное, в SSD используется «wear levelling» [управление износом] – то есть нагрузка равномерно распределяется по «диску» и операции записи не делаются на один и тот же сектор снова и снова.

Я использую журналируемые файловые системы (ext3 и xfs), а также раздел подкачки на моем Eee PC900 уже целый год. Работаю с ним ежедневно; вдобавок там стоит нестабильная сборка Gentoo, и пакеты обновляются почти каждый день. Прибавьте к этому активное использование почты и Интернета (кэш почтовика перезаписывается почти так же интенсивно, как и кэш браузера). Однако ошибки на диске у меня случались только с SD-картой памяти, единственным устройством, не применяющим «wear levelling» (она к тому же была от неизвестного изготовителя и сомнительного качества). А при необходимости спящего режима без раздела подкачки не обойтись.

Вспомните, что нетбуки Eee продаются с двухлетней гарантией, и компании Asus невыгодно включать в свои продукты технологии, способные дать сбой в пределах этого периода. Canonical также не является членом мафии вредителей аппаратного обеспечения (как и любая другая команда разработчиков дистрибутивов); хотя я бы и поставил под вопрос использование *atime* для монтирования файловой системы, но скорее по причинам производительности, чем надежности – я тоже использую *noatime* для жестких дисков. В остальном я вполне одобряю настройки, применяемые по умолчанию в NBR.

3 Удаленный сервер для резервных копий

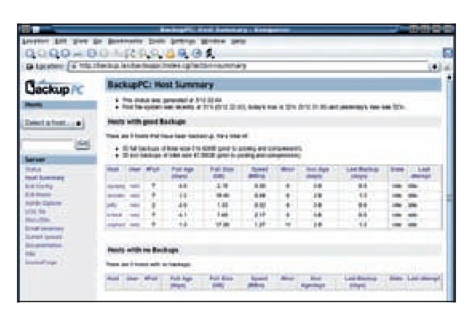
Я планирую настроить удаленный сервер резервного копирования для хранения данных с различных рабочих столов и серверов Windows, и ищу пакет для резервного копирования для Linux-серверов, а также клиентское ПО для управления резервным копированием с клиентской машины.
Джорж Лианос [George Lianos]

Советуем глянуть на *BackupPC* (http://backuppc.sourceforge.net), эта программа предназначена для резервного копирования на базе сервера. Другими словами, на клиентской машине не требуется никакого специального ПО: все процессы резервного копирования запускаются и контролируются сервером. Что, в свою очередь, означает: Вы не будете зависеть от пользователей по части своевременности резервного копирования или заносить задачи в *Cron* на каждом компьютере.

У *BackupPC* есть web-интерфейс, через который доступны просмотр и восстановление копий полностью или в виде отдельных файлов и каталогов. Восстановление можно производить на то же место в файловой системе, где была сделана резервная копия, или загрузить необходимые для восстановления файлы как архив tar или zip. Web-интерфейс предоставляет опцию просмотра всех компьютеров, находящихся в ведении программы, с подробностями обо всех сделанных резервных копиях. В случае возникновения ошибок в результате выполнения копирования, можно даже получать уведомления по почте, но, как правило, работа *BackupPC* не требует вмешательства (лично у меня такая ситуация иногда возникает, когда я выключаю ноутбук во время резервного копирования).

BackupPC связывается с клиентами через *Samba*, SSH, NFS или *rsync*, так что их машины реформировать незачем – просто убедитесь, что у *BackupPC* есть права на чтение разделяемых ресурсов или на подключение по SSH. Не исключено, что резервные копии Вам требуются для нескольких аналогичных машин; тогда Вам будет приятно узнать, что *BackupPC* экономит время и место на диске, храня множественные копии одного файла в виде жестких ссылок. Если на десяти машинах содержится один и тот же файл, сервер сохранит лишь одну его копию.

Здесь нет места для подробных инструкций по настройке BackupPC, но у программы обширная документация, и через пару месяцев мы опишем настройку сервера резервного копирования в одном из наших учебников серии «Сети».



› ***BackupPC* справляется с резервным копированием для многих машин без установки на них дополнительных программ.**

4 Помощь в документации

В апрельском номере [LXF117] вы дали очень экономное решение проблемы прав доступа для файла .dmrc. У меня приключилась такая же проблема, и ваши две строчки ликвидировали ее быстро и просто:

```
chown -R user: ~user
chmod -R u+rw,go-w ~user
```

Я не программист, но мне тоже хотелось бы поучаствовать в улучшении удобства для пользователя и изменить сообщение об ошибке так, чтобы в него включалось это возможное решение. Как мне это сделать?
Джон Стайлз [John Stiles]

Область, где пользователи всегда могут внести вклад в открытое ПО, даже не имея опыта в программировании – это повышение качества документации. Есть два пути предложить свою помощь и предоставить улучшения. Менее формальный способ – сделать сообщение в списке рассылки или на форуме (почти у каждого проекта есть свой форум) и описать свои предложения. Разработчики могут и не отреагировать, но в любом случае обсуждение других пользователей гарантировано.

Другая возможность – отослать сообщение об ошибке на баг-трекер; такие есть у большинства проектов. Несмотря на свое название, баг-трекеры не только отлавливают ошибки (баги): на них также можно высылать усовершенствования и просьбы о введении дополнительных функций. Все запросы отслеживаются, так что и Вам, и разработчикам будет сразу видно, когда были приняты меры по Вашему запросу или сообщению.

В случае, подобном данному, вполне уместен отчет об «ошибке» с соответствующей заменой текста сообщения, но постарайтесь быть кратким. Если необходим более подробный текст, лучше выложить его куда-нибудь в сеть, и пусть сообщение об ошибке отправляет пользователя к нему. Если Вы можете предоставить заплатку на исходные коды программы, будет еще лучше.

Для исправления ошибок в документации лучшим методом будет скачать свежий tar-архив с исходным кодом и с помощью *grep* отловить файл, содержащий сообщение, которое Вы хотите исправить. Создать заплатку очень просто. Скажем, нам нужно изменить файл **errors.h** в текущем каталоге. Делаем его копию под названием **error.h.orig**, а затем вносим необходимые изменения в **errors.h**. Заплатка готова:

```
diff -u errors.h.orig errors.h >errors.h.patch
```

Прочитав файл заплатки, Вы увидите, что это набор инструкций для команды patch: какие строки убрать из старого файла и какие добавить, чтобы он превратился в новый файл. Отошлите заплатку вместе с сообщением об ошибке, и разработчики воспроизведут Ваши изменения. Не отсылайте новый полученный файл целиком, потому что у разработчиков могут быть и свои изменения. Файл заплатки даст им возможность внести Ваши поправки в новую версию программы. »

5 USB так USB

Я пользуюсь компьютером с процессором AMD 1,6 ГГц, на нем установлен Ubuntu 8.10, а на другом разделе — Windows XP. Мой провайдер — BT, и я пользуюсь USB-модемом BT Voyager 105. Тут-то и начинаются проблемы, поскольку в BT сказали, что ничего про Ubuntu не знают и не могут помочь мне настроить подключение к сети. Я скачал какие-то заумные инструкции с форума Ubuntu, но они не помогли. Наверняка существует более простой, пошаговый способ настройки Интернета в Ubuntu.

Кстати, мой принтер Lexmark тоже не работал, и мне пришлось раскошелиться на HP 4100. В связи с этим не советуйте мне купить дополнительное ПО: я пенсионер и сейчас на мели!

Грэм Филлипс-Льюис [Graham Phillips-Lewis]

Понятно, что Вам не хочется зря тратить деньги, но умеренное вложение в более достойный модем спасло бы Вас от множества проблем. Лучшее, что можно сказать про бесплатные модемы, раздаваемые провайдерами – то, что они соответствуют своей стоимости. Порядочный модем подключается к порту Ethernet, а не к USB, и ему не нужны ни специальный драйвер, ни дополнительное ПО на компьютере. Стандартный сетевой стек и web-браузер, которые есть у всех, прекрасно подойдут. Большинство внешних модемов также включают функции маршрутизатора и межсетевого экрана, защищая Вашу систему независимо от установленной ОС. Поскольку хороший модем сам регулирует сетевые протоколы, не перекладывая эту работу на процессор с помощью драйвера, то и сетевое соединение, и сам сам компьютер в общем становятся более отзывчивыми. Подобный модем стоит где-то в районе пары тысяч рублей.

Если Вы все-таки намерены оставить Voyager USB, придется смириться с тем, что он будет работать с меньшей отдачей и что для его настройки придется немного попотеть. Это касается и Windows, но там есть автоматическая установка драйверов. Чтобы проделать это в Ubuntu, надо будет скачать два файла. Так как в Linux Ваша сеть пока не работает, сделайте это из-под Windows или с другого компьютера. Зайдите по адресу http://eciadsl.flashtux.org/download.php и заберите пакет для Ubuntu – текущая версия **eciadslusermode_0.12-1_i386.deb**. Затем на сайте http://archive.ubuntu.com/ubuntu/pool/universe/r/rp-pppoe возьмите свежий Deb-пакет для архитектуры i386. На данный момент это **pppoe_3.8-3_i386.deb**. Номер версии в этих пакетах может быть другим, если следующие релизы выйдут уже после написания этого ответа. Скопируйте скачанные файлы на USB-брелок и перенесите на свой Ubuntu. Убедившись, что модем не подключен, двойным щелчком установите каждый из этих пакетов, сначала пакет с *pppoe*.

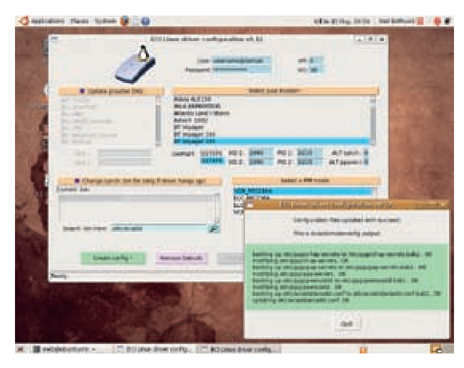
Теперь настройте модем на работу со своим провайдером (для пользователей Windows эта часть обычно выполняется автоматически). Наберите в терминале

```
sudo eciadsl-config-tk
```

чтобы запустить графическую программу настройки (если она не запустится, наберите **eciadsl-config-text**). В верхней части окна введите имя пользователя и пароль, выданные Вашим провайдером, и укажите значения 0 и 38 для VPI и VCI соответственно. Выберите нужный модем из списка, выставьте режим PPP на VCM_RFC2364, щелкните по кнопке Remove Dabusb (игнорируя любые возможные сообщения), а затем – по Create Config. В результате должно открыться диалоговое окно с сообщениями, заканчивающимися на OK. Теперь модем установлен и настроен; включите его и войдите в Интернет командой

```
eciadsl-start
```

Ее можно привязать к иконке на рабочем столе, щелкнув по нему правой кнопкой мыши и выбрав в меню пункт Create Launcher [Создать кнопку запуска]. Внесите eciadsl-start (или eciadsl-stop) в поля имени и команды. Также можно заставить стартовать соединение автоматически при запуске рабочего стола с помощью меню System > Preferences > Sessions [Система > Настройки > Сессии], щелкнув там на кнопку Add [Добавить] и вписав имя программы.



› **Настройка USB-модема Voyager — сущие танцы с бубном, но раз нет Ethernet ADSL-модема, то эти усилия оправданы.**

6 Не-переносимость

Что творится с принципом переносимости в Linux-сообществе? Я знаю, что разные дистрибутивы обязаны работать на разных машинах с разными конфигурациями. Но ведь этот принцип должен применяться и к обновлениям одного дистрибутива! У меня двухпроцессорная 64-битная машина Acer Aspire с графической картой GeForce 8200, прекрасно работающей в Fedora 9 (признаю, я-таки повозился с разрешением экрана, но это ерунда). А вот когда я пытаюсь обновиться до 64-битной Fedora 10, *X Window System* не стартует, работает только консольный вход, и startx ничего не делает. Почему *X*-сервер в Fedora 9 работает, а в Fedora 10 — нет? Разве не должны одни и те же программы работать в той же ОС, хотя бы и обновленной?

Виллим Браун [Willim Brown]

Ответ на Ваш вопрос – «да, должны», но, боюсь, Вам будет этого мало. Сомневаюсь, что **startx** совсем ничего не делает. Возможно, он не запускает *X*, но в любом случае выдает в терминал сообщения об ошибках. Более подробную информацию можно найти в фай-

Вопрос-победитель (английская версия)

Клонирование CD

Пытаюсь собрать компьютер для клонирования CD на базе старенькой машины Sempron с четырьмя приводами IDE CD-RW и жестким диском SATA. Я установил Ubuntu 9.04 и попробовал *GnomeBaker* и *K3b*, но в них, похоже, нет поддержки множественного одновременного прожига. Нам нужно готовить около 200 дисков для различных проектов в школе, где я работаю — не посоветуете ли вы что-нибудь подходящее новичку?

Dwillmot20, с форумов

Я бы создал ISO-образ с помощью Вашей любимой программы и применил скрипт оболочки: для повторяемой задачи это куда удобнее, чем все время жать на кнопки в графическом интерфейсе. Например, так:

```
#!/bin/sh
DEVICES="/dev/cdrom0 /dev/
cdrom1 /dev/cdrom2 /dev/
cdrom3"
for DEV in DEVICES; do
cdrecord -eject dev=$DEV "$1" &
done
```

Скопируйте это в любой текстовый редактор (допустим, в Gedit), перечислите свои записывающие CD-приводы в строке DEVICES, сохраните скрипт в каком-либо каталоге, внесенном в PATH – скажем, в **/usr/local/bin/multiburn.sh** или **~/bin/multiburn.sh** – и сделайте его исполняемым. При сохранении в домашнем каталоге **bin** это поможет сделать файловый менеджер: щелкните правой кнопкой по файлу и выберите Свойства. Или установите права в консоли:

```
sudo chmod +x /usr/local/bin/
multiburn.sh
```

Создав ISO-образ с помощью *K3b* или любой симпатичной Вам программы, вложите чистые болванки во все приводы и запустите скрипт, задав ему путь к ISO-образу.

```
multiburn.sh /path/to/image.iso
```

Когда все четыре диска будут записаны и извлечены, замените их и запустите скрипт снова. Скрипт можно модифицировать, чтобы он запускался по нажатию указанной клавиши. Правда, для повторного запуска скрипта нужно нажать всего две клавиши (стрелка вверх и Enter), так что времени это не сэкономит, но зато станет хорошим практическим упражнением, если Вы расположены чему-нибудь научиться.

ле журнала **/var/log/Xorg.0.log**. Ошибки помечены как "(EE)", и выделить их из остального информации в этом файле можно с помощью

```
grep EE /var/log/Xorg.0.log
```

В некотором смысле дистрибутивы находятся на милости разработчиков ПО. И Вас подвели изменения в *X.org*. Команда разработчиков *X.org* старается перейти от громоздкого и иногда загадочного **xorg.conf** к системе полного автоопределения, чтобы любая аппаратная комбинация сразу работала сама, но в Вашем случае так не получилось. Вот самое быстрое решение проблемы: если у Вас сохранилась установка Fedora 9 или резервные копии важных файлов, скопируйте **/etc/X11/xorg.conf** из установки Fedora 9 в Fedora 10. *X.org* предпочтет содержимое этого файла автонастройкам, и все должно заработать.

Если у Вас нет файла **xorg.conf**, его можно создать для Вашей аппаратной конфигурации, запустив из-под root следующие команды:

```
yum install system-config-display
system-config-display
```

В результате должно появиться то самое окно настройки, которое Вы видели во время установки Fedora 9, и на основе Вашего выбора создастся **/etc/X11/xorg.conf**. Если и тогда ничего не выйдет – скажем, не обнаружатся подходящие настройки для Вашего оборудования – укажите нужные опции из командной строки, к примеру,

```
system-config-display --setresolution=1024x768
--set-depth=24 --setdriver=nvidia
```

Запустите **system-config-display --help**, чтобы увидеть все доступные опции.

7 Просто Sed

Я пишу *Bash*-скрипт, который, среди прочего, устанавливает всякие программные пакеты. С помощью *Zenity* я придал ему привлекательную графическую оболочку и запускаю его на 64-битном Debian 5.0.

К сожалению, если один из пакетов, который мне нужно установить, находится на установочном DVD с Lenny, возникает проблема: в консоли под графической оболочкой появляется сообщение «Пожалуйста, вставьте DVD», но *Zenity* его не видит и не передает. А пользователю кажется, что программа зависла. Соответствующий фрагмент кода выглядит так:

```
Select all
( echo "33"
aptitude -y install gcc
echo "66"
aptitude -y install sysstat
echo "99"
) | zenity --auto-close --progress
--text="Fetching software..." --title="Installing
Software " --width 300
```

Тут возникает ряд вопросов. Во-первых, можно ли заставить *Zenity* распознавать события и показывать сообщения типа «пожалуйста, вставьте установочный DVD»? Думаю, что нет, хотя могу ошибаться. Но это было бы идеальным решением, и тогда мой вопрос даже не нужно дочитывать до конца (только подскажите нужное заклинание!).

Как вариант, я мог бы написать код, говорящий: «Если установочный DVD занесен в /etc/apt/sources.list, закомментируй это» и «Если в источниках нет http://volatile.debian.org/debian-volatile — добавь его». К сожалению, для анализа текстовых комментариев файлов придется привлечь *awk* и *sed*, а оба они не вызывают у меня энтузиазма! Я совершенно беспомощен в работе с ними, хотя и штудировал их man-страницы и статьи про них в Google до рези в глазах.

Не посоветуете ли вы код для *Bash*, который проверял бы присутствие двух источников в файле *sources.list*, удалял строчки насчет DVD при их наличии, добавлял бы онлайн-источник, если он там отсутствует, и не добавлял бы его, если он уже записан в файле?!

HJR, с форумов

Посылается ли сообщение «Пожалуйста, вставьте...» на стандартный вывод ошибок вместо стандартного вывода сообщений? Если да – его можно отловить, добавив параметр 2>&1 к Вашему вызову **aptitude**. Затем просмотрите вывод в поисках нужной строки, прежде чем передавать его *Zenity*.

Вы правы, предполагая, что для изменения текстовых файлов из командной строки нужны *sed* или *awk* – в данном случае *sed* – но эти утилиты достойны изучения. Поначалу они вводят в трепет, но когда Вы освоитесь, станут для Вас незаменимыми. Все источники CD/DVD в файле **sources.list** закомментирует такая команда:

```
sed -i 's/^deb cdrom/# deb cdrom/' /etc/apt/sources.list
```

Разберемся: **-i** значит «заменить существующий файл измененным содержимым»; **s** значит «вме- »



Часто задаваемые вопросы

Удаленные рабочие столы

» Хочу подключаться к моему домашнему компьютеру с работы и запускать графическую программу, например, почтовый клиент; возможно ли это?

Да, возможно. Существует два основных способа этого достичь, с различными вариантами в каждом случае.

» Что это за различные варианты?

Первый – *X*-проброс (*X*-forwarding), который лучше подходит, если на обоих компьютерах установлен Linux. Он работает по SSH-соединению: вы запускаете программу в удаленной оболочке, и она показывается на вашем локальном рабочем столе. Например:

```
ssh -X me@my.home.computer
kmail
```

Опция **-X** велит SSH использовать *X*-проброс.

» Это все, что нужно для запуска программ на моем локальном компьютере?

Строго говоря, программа запускается на удаленном компьютере, но ее окошко показывается в локальной системе. Возможно, вам понадобиться отредактировать файл **/etc/ssh/sshd_config** и выставить значение 'yes' напротив опции 'X11Forwarding'.

» А вдруг мне понадобится весь рабочий стол — справится ли с этим *X*-проброс?

Да, справится. При этом запустится новый графический сеанс, который будет показан на вашем локальном компьютере.

» Но ведь это не одно и то же! Если я оставил программу работать дома и хочу увидеть ее окно с работы, у меня ничего не получится?

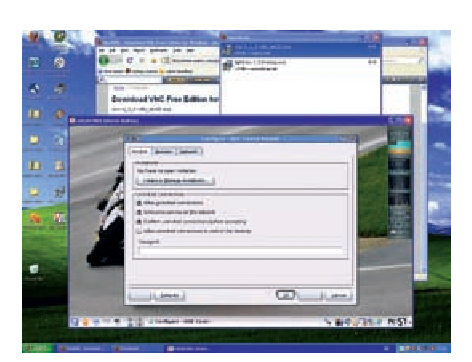
Нет, для этого понадобится программа удаленного рабочего стола, так же, как если бы на вашем локальном компьютере стояла Windows. Стандартным выбором тут будет VNC (Virtual Network Computer). Это система «сервер–клиент», где VNC-сервер запущен на вашем удаленном компьютере, а клиент – на локальном. На сайте www.realvnc.com есть пакеты для всех популярных ОС, так что вы можете попасть на вашу Linux-машину из системы Windows или Mac, и наоборот.

» Какие программы мне понадобятся?

А какой у вас рабочий стол? Если KDE, то все, что нужно, уже есть, потому что в KDE встроено ПО для удаленного фрейм-буфера (RFB, remote frame buffer). RFB – это протокол, используемый VNC и совместимыми системами. С другими графическими средами вам понадобится установить из репозиториев вашего дистрибутива пакеты VNC или *TightVNC*.

» А как насчет ширины канала? Можно ли подключаться через модем?

Модему не потянуть вывод удаленного графического рабочего стола и поддержку его в реальном времени. Сгодится широкополосное подключение, но учтите, что независимо от входящей скорости вашего ADSL, исходящая вряд ли превышает 384 кбит/с, поэтому лучшим выбором будет *TightVNC*. Помогут уменьшение размера и глубины цвета, отключение графических эффектов и прочие меры по уменьшению потока данных.



› Убедитесь: вот рабочий стол KDE, запущенный из Windows XP через VNC.

сто всего, что совпадает с первой строкой, вставить вторую строку». Так мы заменяем все выражения **deb cdrom** в начале строки на **# deb cdrom**. Точно так же и *Synaptic* изменяет файл, чтобы источник при случае снова мог быть доступен.

Для добавления строки с источником просто запишите ее в файл с помощью *echo*, но файл **sources** лучше зря не трогать; сперва проверьте, надо ли вообще его менять, с помощью *grep*. До внесения правок сделайте резервную копию файла и восстановите его, когда все закончите.

```
cp /etc/apt/sources.list /etc/apt/sources.list.$$
sed -i 's/^deb cdrom/# deb cdrom/' /etc/apt/
sources.list
grep -q "^deb http://volatile.debian.org/
debianvolatile"
/etc/apt/sources.list || echo
"строка репозитория" >>/etc/apt/sources.list
# делайте свои дела
if diff -q /etc/apt/sources.list /etc/apt/sources.
list.$$
then
rm /etc/apt/sources.list
else
mv /etc/apt/sources.list.$$ /etc/apt/sources.
list
fi
```

Здесь **$$** – номер текущего процесса, он обеспечит уникальность имени файла для резервной копии. Команда **grep** означает, что репозиторий добавится только в том случае, если его не хватает, а заключительная часть проверяет, были ли сделаны изменения в **sources.list**, и восстанавливает его первоначальный вариант, если файл был изменен.

8 Молчит, как партизан

В После обновления до Debian Lenny у меня появились странные проблемы со звуком: при воспроизведении аудио и видео он присутствует, а при проигрывании Flash-роликов в браузере — нет. Также наблюдаются проблемы со звуком в *VirualBox* и *Wine*. При этом в консоли появляется сообщение:

```
ALSA lib pcm_dmix.c:996:(snd_pcm_dmix_open)
unable to open slave
```

Запуск *alsaconf* от имени root решает проблему, и звук появляется во всех приложениях, но исчезает микрофон, встроенный в web-камеру Logitech. Во время перезагрузки компьютер жалуется, что по адресу index=0 аудиокарты не обнаружено. После полной загрузки снова работает микрофон, но вообще нет звука.

Заменив в файле /etc/modprobe.d/sound **значение index на 1 и перегрузившись, я возвращаюсь к исходной точке: нет звука в браузерах и *VirtualBox*, но есть микрофон. Нельзя ли настроить и его, и звук одновременно? На всякий случай, привожу содержимое** /etc/modprobe.d/sound**:**

```
alias snd-card-0 snd-hda-intel
options snd-hda-intel index=1
```

Михаил Михайлов

О Когда в системе более одного аудиоустройства, звуковая подсистема *ALSA* различает устройства по номерам: **hw:0**, **hw:1** и т.д. Приложения обычно берут устройство по умолчанию, первое в этом списке, то есть **hw:0**. Чаще всего звуковое устройство в компьютере только одно (аудиокарта), и настройка приложений, т.е. указание, через какое устройство воспроизводить звук, не требуется.



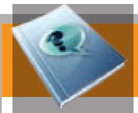
› **Программа настройки экрана в Fedora запустится и без *X*-сервера, и это удобно, поскольку чаще всего она нужна, когда *X*-сервер не работает.**

В Вашем случае, встроенный в видеокамеру микрофон стал для системы дополнительным звуковым устройством. Если оно первое в списке, то приложения пытаются вывести звук через него, но это всего лишь web-камера, и ничего хорошего тут не выйдет.

Чтобы разрешить конфликт между микрофоном и звуковой картой, жестко задайте очередность их загрузки при старте ОС. Для этого в файл **/etc/modprobe.d/sound**, содержимое которого Вы привели, достаточно добавить строки:

```
alias snd-card-0 snd-hda-intel
alias snd-card-1 snd-usb-audio
options snd-hda-intel index=0
options snd-usb-audio index=1
```

Здесь мы явно указываем, какое устройство должно быть первым (здесь это **snd-hda-intel**), а какое – вторым (**snd-usb-audio** – web-камера, подключенная через USB).



Краткая справка про...

Initrd

Заглянув в свой каталог /boot **или в конфигурационный файл меню загрузчика, вы увидите ссылки на файлы initrd. Это образы электронного диска ramdisk (или образы ramfs, новой реализации того же принципа). Ramdisk, как ясно по названию, это дисковое накопительное устройство, целиком расположенное в ОЗУ компьютера. А файл initrd — это ramdisk, сохраненный в файл. Теперь, зная, ЧТО это, давайте разберемся, ЗАЧЕМ оно.**

Initrd, или начальный ramdisk (initial ramdisk) – это электронный диск, загружаемый ядром при старте. Он становится корневой файловой системой, и оттуда запускаются скрипты настройки системы перед передачей контроля настоящему корневому разделу на жестком диске. Главная задача электронного диска – загрузка модулей ядра. Дистрибутивы созданы для работы на самом различном оборудовании, значит, собирается множество драйверов устройств. Если модули вкомпилировать в образ ядра, оно станет гигантским, тормозным и съедающим прорву памяти, а 90 % его содержимого окажется мертвым грузом. Поэтому в Linux созданы загружаемые модули ядра. Модули, нужные для монтирования корневой файловой системы, очевидно, не могут быть считаны из нее, и нужен способ заранее загрузить драйверы для контроллера жесткого диска и файловой системы, и, возможно, некоторые другие, например, *LVM* или *dm-crypt*. Этим и занимается ramdisk. Скрипт *linuxrc* с него загружает модули, запускает все необходимые программы настройки (например, для *LVM* или зашифрованной корневой файловой системы), а затем переходит на корень жесткого диска.

Образ initrd – это файловая система в файле cpio, сжатом с помощью *gzip*, так что его можно распаковать, смонтировать, изменить и запаковать снова таким образом:

```
cd /mnt/tmp
zcat /path/to/initrd | cpio -id
#modify files here
find . -depth | cpio -o | gzip >/path/to/
newinitrd
```

9 Как в кино

В Пожалуйста, посоветуйте, как программно отключить цветность *X*-сервера в Linux: хочу получить полностью черно-белое изображение с максимумом градаций серого, как в старом добром кино. Еще лучше было бы задать какой-нибудь тон, вроде сепии.

Это меня очень здорово выручило бы. Мне приходится подолгу сидеть за монитором, а цвет для моей работы, в общем, не нужен, и я заметил, что когда убираю его аппаратно, глаза устают меньше.

Александр Винокуров

О Исходя из личного опыта, я бы советовал сменить монитор, но это не есть решение Вашей задачи. Обойдемся без финансовых затрат. В секции Display файла конфигурации *X*-сервера (обычно **/etc/X11/xorg.conf**) имеется параметр Visual; он может принимать значения StaticGray и GrayScale. Укажите одно из них, и Ваша проблема будет решена.

Но учтите, черно-белый *X* доступен только при глубине цвета до 8 бит, и Вам придется заодно поменять и параметр DefaultColorDepth в секции Screen. Не помешает также установить на рабочем столе визуальную тему с минимумом цветов: градиенты и прочие новомодные украшения не особо красиво смотрятся даже при 256 оттенках серого.

Большой вопрос Как мне вернуть мой Ubuntu?

10 Починка загрузчика

В Я недавно установила Ubuntu 9.04, но затем по ошибке инсталлировала на другой раздел Windows XP, забыв, что она затирает MBR, и теперь мне недоступна моя почта в Ubuntu.

Потом я подумала, что если поставить Ubuntu еще раз, то, возможно, у меня будет доступ к MBR, а то и доступ к первой установке Ubuntu через измененный MBR.

Могу ли я с помощью второй установки Ubuntu получить доступ к почтовым папкам в моем первом Ubuntu? Или посоветуйте мне, как возобновить к нему доступ.

Патриция [Patricia]

О В MBR хранится только начальный код загрузчика; файлы, нужные для показа загрузочного меню и загрузки по выбору, хранятся на загрузочном разделе Вашей первой установки Ubuntu. Восстановить *Grub* в MBR можно с помощью большинства LiveCD, и некоторые из них даже специально предназначены для этой задачи: например, SuperGrub Disc (www.supergrubdisk.org) или System Rescue CD (www.sysresccd.org). Super Grub Disc может автоматически поправить MBR в случае, если на компьютере одна установка Linux и одна установка Windows. Даже если установлены два Linux, нужно только следовать пунктам меню и выбрать дистрибутив, в котором Вы хотите настроить загрузчик. Подробные инструкции см. на сайте www.supergrubdisk.org/wiki/Howto_Fix_Grub.

Восстановить *Grub* можно с помощью любого Live-CD и нескольких команд. Сначала определите, на каком диске находится каталог **/boot**. *Grub* начинает отсчет дисков и разделов с нуля, так что второй раздел на первом диске, то есть, в терминах Linux, **/dev/sda2**, будет называться (hd0,1). Определив диск, открываем терминал и вводим:

```
sudo grub
root (hd0,1)
setup (hd0)
quit
```

Первая команда вводит нас в оболочку *Grub* с правами root. Следующая – определяет раздел, где находятся файлы загрузчика; затем мы записываем *Grub* в MBR командой **setup** и выходим из оболочки. Если Вы не уверены в разделе, оболочка *Grub* содержит вспомогательную команду **find**. В большинстве дистрибутивов создается символьная ссылка **/boot/vmlinuz**, указывающая на ядро, поэтому после входа в оболочку *Grub* запустите

```
find /boot/vmlinuz
```

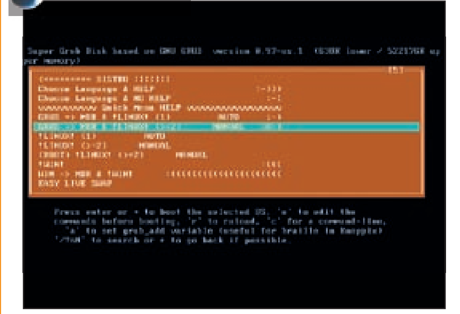
Выдастся по одному результату на каждую из Ваших установок Ubuntu, так что выбирайте нужную.

Если Вам требуется доступ из новой установки в старую, проверьте совпадение ID пользователя и группы. Даже если Вы использовали одно и то же имя пользователя на обеих установках, файловая система хранит только числовой ID, поэтому если у Вашего пользователя был UID 1000 на первой установке и 1001 – на второй, то полного доступа на чтение/запись к файлам из другой установки у Вас не будет. Чтобы сделать их доступными для чтения из текущей установки, введите в консоли

```
chown -R username: /other/home/username
```

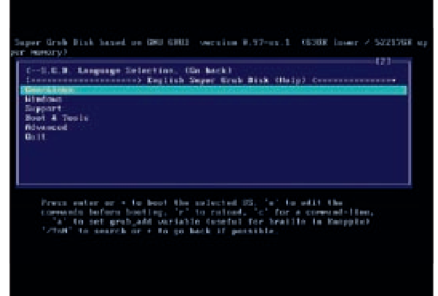
заменив username и точку монтирования на соответствующие значения.

Шаг за шагом: Восстанавливаем MBR



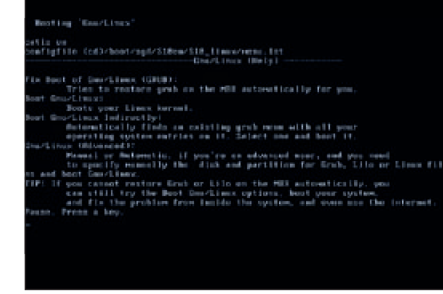
1 Super Grub Disk

Super Grub Disk, установленный на CD или USB-флэшку, загружается прямо в меню, где перечислены опции для поправки вашего MBR и загрузчика.



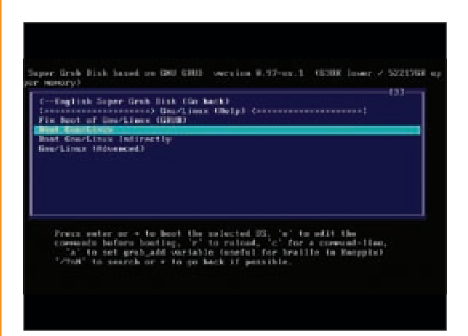
2 Онлайн-помощь

Прежде чем ковыряться в MBR, отведите пару минут на чтение документации про опции, которые вы собрались применить.



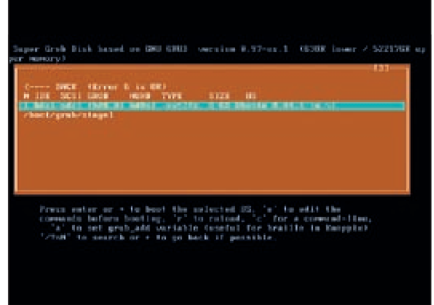
3 Объяснение опций

Меню помощи для каждой опции объясняет, что именно она реализует; нажмите любую клавишу, чтобы перейти прямо к опции.



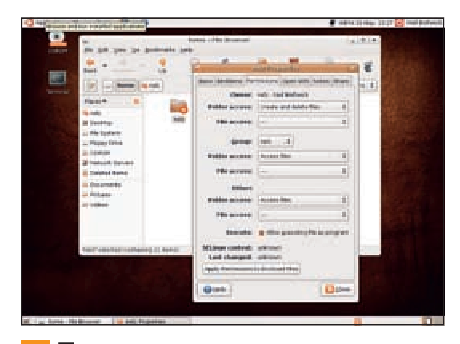
4 Загружаем Linux

Если вы не хотите менять MBR, с помощью Super Grub Disk можно загрузиться в любую из операционных систем, установленных на диске.



5 Исправляем MBR

В случае нескольких установленных систем Linux нужно выбрать ту, в которой хранятся конфигурационные файлы загрузчика.



6 Проверить права

Если вам нужен доступ к двум установленным системам с одинаковыми именем пользователя и паролем, поменяйте их числовые ID командой *chown*. LXF

LXFHotPicks



Энди Хадсон
Когда Энди не притворяется, что отлаживает почтовые сети, он ныряет среди коралловых рифов в поисках жемчужин для HotPicks.

Клиент BitTorrent

Flush

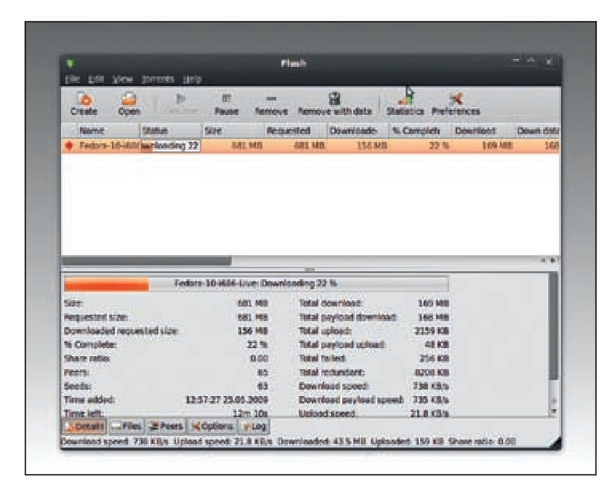
Версия 0.5 **Сайт** http://sourceforge.net/projects/flush

На наших страницах не раз мелькали клиенты BitTorrent, отчего, возможно, мы сделались излишне строги к новичкам. Дело в том, что выпустить на свет средненький клиент, не хуже толпы всех прочих, не составляет особого труда. По счастью, нам повстречался *Flush*, который временно осадил наш цинизм.

Flush обычно запускают, скачав torrent-файл и открыв его в клиенте. После этого вы наблюдаете за процессом загрузки с помощью числовых индикаторов размера файла и процента уже полученных данных. Кнопки внизу экрана помогут добыть более подробные сведения по количеству пиров [peer], к которым подключены вы или которые подключены к вам, а также по IP-адресам и клиентам.

Большую часть этих сведений легко получить в сокращенной форме – нажав на кнопку Statistics, что также позволит сбросить все счетчики, если вы следите за производительностью всерьез. Интерфейс довольно прост, и разработчик явно не поленился влезть в шкуру конечного пользователя.

> «В довершение опций, имеется возможность создать torrent-файл.»



› *Flush* — клиент, облегчающий работу по управлению torrent'ами.

Flush — море опций

Параметров для настройки Flush множество: например, можно указать рабочую и конечную директории, что весьма удобно, если вы хотите допускать пользователей только к завершенным закачкам. Можно также заставить *Flush* прослушивать определенный диапазон портов, есть и подобие автоматизации: Flush просто «подхватит» torrent-файл после его копирования в определенное место. Это хорошо, но могло бы быть еще лучше, если бы время начала закачки настраивалось. В довершение набора опций имеется возможность создания собственного torrent-файла для раздачи. Соответствующий интерфейс отличается прямолинейностью – просто укажите *Flush* файлы для раздачи, добавьте местоположения трекеров и нажмите на Create.

По части зависимостей, этому аккуратному клиенту требуются *libtorrent*, *libboost* и *gtkmm*.

Исследуем интерфейс Flush

Создание
Жмите сюда, чтобы создать собственный torrent.

Состояние
Отображает состояние закачки в числовом виде.

Пиры (Peers)
Как и большинство клиентов, *Flush* показывает, кто сейчас с вами связан.

Значки
Здесь вы получите подробную информацию по вашей работе с torrent'ами.

Удаление
Загрузили не те файлы? Это позволит вам удалить torrent и все его данные.

Статистика
Получите всестороннюю информацию по состоянию закачки.

Инструмент настройки

Ubuntu Tweak

Версия 0.4.7 **Сайт** http://ubuntu-tweak.com

Одна из заведомо сильных сторон Ubuntu – относительная простота в использовании, и пока остальные дистрибутивы играли в догонялки, Ubuntu созрел как самая популярная из ныне существующих разновидностей Linux. Согласно этой позиции, тем больше внимания уделялось созданию инструментов и утилит, помогающих пользователю настроить систему по своему вкусу, и именно в таком духе мы случайно наткнулись на совершенно правильно названный *Ubuntu Tweak* [tweak означает «настройка», – *прим. пер.*].

Это пакет, созданный для обеспечения прямого доступа к настройкам, за которыми обычно приходится буквально охотиться, во многом похожий на то, что делал *TweakUI* для Windows. Охватывая широкий диапазон параметров – например, управление запуском, рабочим столом и персонализацией, *Ubuntu Tweak* дает отличный способ настроить Ubuntu согласно вашим личным пристрастиям, не тратя долгих часов на поиск нужной информации в Google. Он даже предоставляет доступ к некоторым наиболее популярным альтернативным репозиториям, например, Google's Linux Repository и репозиторию *Opera*.



› Хотите донырнуть до дна всех функций Ubuntu? Тогда, возможно, *Ubuntu Tweak* — именно ваш инструмент.

Чистка системы

Установив программу, вы найдете ее в разделе Системные [System Tools] меню Приложения [Applications]. Откройте ее, и вас поприветствует экран краткого введения, где рассказывается о функциях программы. Каждая группа слева в главном окне раскрывается, предоставляя вам доступ к опциям. Например, в разделе Applications имеются Add/Remove [Добавить/удалить], Source Editor [Редактор источников], Third Party Software [Сторонние приложения] и довольно полезный Package Cleaner [Чистильщик пакетов]: он поможет вернуть в оборот дисковую память, забитую ненужными пакетами и кэшем – раньше тут было не обойтись без командной строки. *Ubuntu Tweak* оперирует внутри пользовательского каталога, но способен вносить изменения и во всю систему, через *policykit*.

> «Полезный Package Cleaner вернет вам дисковую память.»

Ubuntu Tweak существует в виде Deb для Ubuntu или в виде tar-архивов, и подойдет не всем – в частности, отпадают пользователи Fedora; но все же его ждет немало поклонников среди новых (и не очень) пользователей Linux, только что перешедших на Ubuntu.

Музыкальный менеджер

Cactus Jukebox

Версия 0.4.1 **Сайт** http://cactus.hey-you-freaks.de

Наши личные фонотеки постоянно растут, а с ними растет и важность наличия достойных программ по управлению ими. *Cactus Jukebox* – еще один претендент на это звание, хотя его интерфейс сперва кажется перегруженным.

Тем не менее, начинаете вы с пустой библиотеки – *Cactus* не вынуждает вас держать все музыкальные файлы в директории Music, которая сейчас имеется в большинстве дистрибутивов на базе Gnome. Заглянув в меню File [Файл], вы увидите пункт Add Directory [Добавить директорию], позволяющий просматривать папки с музыкой до самого верхнего уровня и сканировать каталоги более низких уровней на предмет составления списка всех совместимых файлов – поддержка OGG есть по умолчанию. Слева помещен список имеющихся исполнителей, а с каждым из них связан перечень его доступных альбомов. Для просмотра можно использовать дерево директорий, а сами треки появляются в панели над значками управления. *Cactus* также предусматривает быструю и удобную функцию поиска, применяющую нечеткую логику и позволяющую найти трек по его названию, названию альбома или имени исполнителя.



› Универсальный инструмент управления фонотекой, *Cactus Jukebox* умеет все — от и до.

Вкладки управления

В дальней левой стороне находятся три разные вкладки: первая – для вашей музыкальной библиотеки; вторая – для работы в сети, например, для потокового интернет-радио; а третья позволяет управлять музыкой на медиа-плейере, перетаскивая в него треки.

Если вы намерены использовать *Cactus* для создания CD – чтобы помочь в составлении диска, понадобится пакет *cdda2wav*; кроме того, проверьте наличие у вас установленного кодировщика MP3 и *MPlayer*.

Cactus – полезный проект, избравший свой путь, а не дорогу, прототоренную *RhythmBox* и *Banshee*; в частности, потому, что полагается на одного-единственного разработчика. Мы считаем, что он заслуживает большего интереса широкого Linux-сообщества, и приглашаем вас познакомиться с ним и протянуть руку помощи; мы уверены, что дополнительная поддержка его очень украсит.

Системный планировщик задач

Gnome Schedule

Версия 2.10 **Сайт** http://gnome-schedule.sourceforge.net

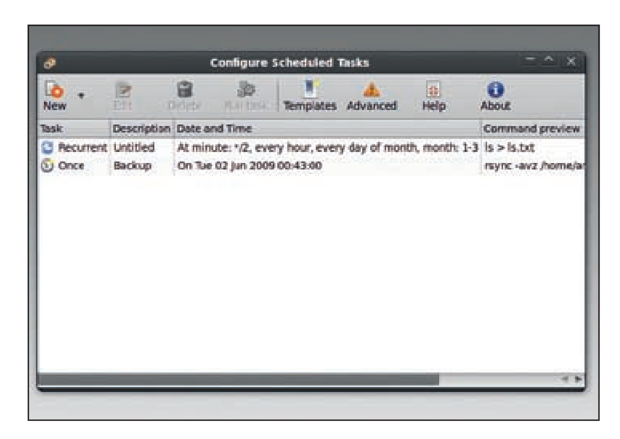
Если и есть нечто общее у всех системных администраторов, так это склонность к автоматизации мелких задач, ради возможности заниматься делами поважнее. Большинство из них знакомы с crontab, хранилищем инструкций, используемых *Cron* для выполнения задач согласно заданному плану, но дома вам скорее всего понадобится запланировать разве что регулярное резервное копирование или периодический запуск скрипта. Для подобного сценария нет нужды зарываться в командную строку – ведь существует удобная утилита *Gnome Schedule*: она даст вам не только власть над *Cron*, но и ограниченный доступ к команде **at**.

Плановые задачи

Для начала надо убедиться в наличии привязок Python GTK и Python Gnome, а затем заняться компиляцией, используя тройку **./configure**, **make** и **make install** (от имени root). После этого вы обнаружите *Gnome Schedule* угнездившимся среди пунктов раздела Системные [System Tools] в меню Gnome. По его запуску появится простоватый экран, но вы не переживайте – это только холст, на котором вам предстоит работать. Теперь нажмите на кнопку New [Новый], и вам предоставят три типа задач. На данный момент у вас нет никаких шаблонов (их надо будет создать – но об этом позже), так что придется выбрать либо повторяющуюся, либо однократную задачу. В первом случае перед вами появится простое диалоговое окно, где надо будет описать задачу и команду (или скрипт оболочки), подлежащую выполнению, и всплывающее окно, позволяющее контролировать работу команды. При желании запланировать приложение на основе *X* можно также перевести работу *Gnome Schedule* в фоновый режим, подавив его видимую активность.

Далее вы можете настроить основные параметры таймера, включая выполнение задачи при перезагрузке (это удобно, например, для затирания данных, которые нежелательно оставлять после себя). Кроме того, можно создать более подробное расписание, используя область Advanced [Дополнительно]. Нас особо впечатлила глубина детализации, предоставляемая кнопками Edit [Правка]. Они располагаются по сторонам каждого поля Advanced, и с их помощью частота выполнения задач настраивается намного точнее: например, можно указать, что задача должна выполняться в определенную минуту, час или день. Внизу диалогового окна *Gnome Schedule* нормальным языком – что весьма гуманно – сообщит вам, по какому расписанию работает задача, и вы сможете либо выполнить ее, либо добавить как шаблон.

> «Особо впечатлила глубина детализации по кнопкам Edit.»



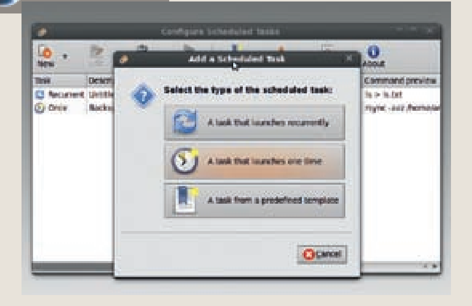
› Пускай *Gnome Schedule* не блещет красотой, зато он многогранен и прост для понимания.

Однократные задачи

Ну, а можно создать разовую задачу, которая выполнится в определенное время определенного дня. Это диалоговое окно намного проще: ведь здесь надо всего лишь указать задачу и дату и время ее выполнения. В нем есть календарь, позволяющий быстро просматривать недели и месяцы. И опять же, можно сохранить задачу в виде шаблона, чтобы его часто использовать – *Gnome Schedule* позволяет накопить серьезную библиотеку шаблонов, и если вы обнаружите, что некая задача хорошо сработала, а то и просто не захотите терять установку даты и времени, то, уж конечно, не упустите шанс превратить все это в шаблон.

В итоге, мы полагаем, что это приложение очень удобно для домашних пользователей, но вряд ли устроит опытных системных администраторов.

Шаг за шагом: Планируем задачу



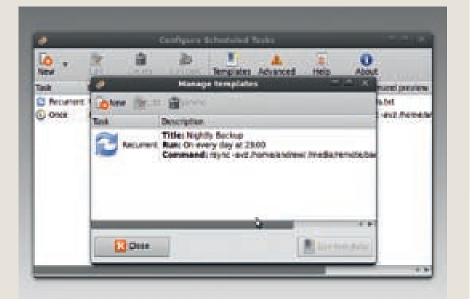
» Начало процесса

Нажав кнопку New [Новый], выбирайте: будет ли это повторяющаяся или однократная задача?



» Отладка

Очень важно правильно настроить команду. Теперь аккуратно составьте расписание для задачи.



» На плечах гигантов

Обретя привычку создавать шаблоны из текущей задачи, вы заведете библиотеку на будущее.

HotGames Развлекательные приложения

Аркада

Atomic Worm

Версия б/н **Сайт** www.charliedoggames.com

Наверное, первой компьютерной игрой, насколько мы в силах припомнить, была *Snake*. Для тех, кто не застал восьмидесятых и не покупал раннюю версию мобильного телефона Nokia, объясняем, что игра основана на очень простой концепции – надо протащить свою змею по экрану, поедая по дороге угощения. Чем больше змея ест, тем длиннее она становится, но надо следить, чтобы она не врезалась в собственный хвост – тут игре и конец.

Оригинал затягивал вас на несколько часов, настолько это было увлекательно. Проблема заключалась в его простоте; по достижении определенных высот, вы теряли интерес к игре и к ней уже не возвращались.

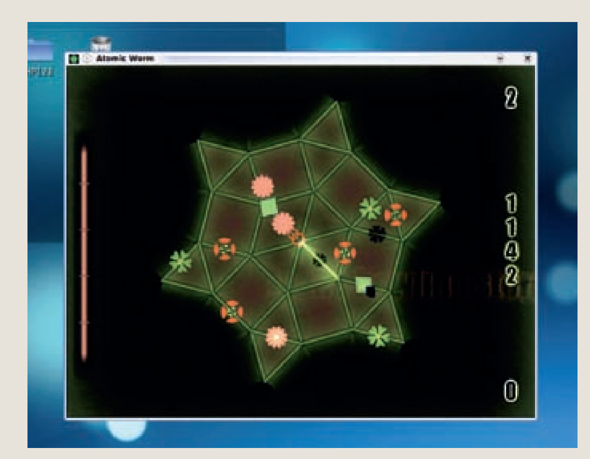
Atomic Worm берет эту достославную игру, придает ей лоск XXI века, и – Боже мой! – как далеко мы ушли за 20 лет. Вместо полной свободы выбора, куда вести змею, вам предлагается перемещать червяка в пределах сетки, показывая мышью, куда ему идти. Разрешается делать только один ход за раз, так что выполнить несколько движений, чтобы выбраться из затора, не получится.

К счастью, червяк всегда движется с одинаковой скоростью по всему пути следования, так что вам надо просто все время двигаться, пока не найдется выход. Форма сетки разная на разных уровнях – вы будете играть в шестиугольниках, треугольниках, квадратах и т. п. – и не всегда фиксированная.

Изменилась не только навигация: вы обнаружите, что разработчики решили добавить мины. Зато, собрав три одинаковых предмета подряд, вы получите замедление скорости на короткое время и щит, помогающий прорваться сквозь мины, а червяк чуть-чуть укоротится. Эти новшества означают, что перед вами – коварная игрушка, освоение которой с гарантией отнимет у вас уйму времени; правда, если вы подвержены укачиванию, движение уровней в ней, возможно, вызовет у вас тошноту. Но это имеет свои плюсы – тогда вам придется иногда отрываться от игры.

Если вы решитесь завить веревочкой *Atomic Worm*, то файл **install.sh** выполнит за вас всю черную работу, так что не беспокойтесь насчет зависимостей.

> «Коварная игрушка с гарантией отнимет уйму времени.»



› Ваш червяк вечно голоден – даже глотая изотопы. Какой вам еще поэзии полета?

Гонки

LINUX FORMAT HotPicks *повторный визит*

SuperTuxKart

Версия 0.6.1 **Сайт** http://tinyurl.com/nn4yqv

Было бы несправедливо обойти вниманием новый релиз легендарной Linux-игры. *SuperTuxKart* долгое время занимал особое место в наших сердцах благодаря ностальгическим воспоминаниям об игре, послужившей для него основой – *Super Mario Kart*.

Суть проста: вы перемещаетесь по трехмерной трассе, пытаясь опередить соперников и прийти к финишу первым, благодаря умелому продвижению по дороге и хитроумному использованию подобранных по пути пакетов. В них – набор оружия для нападения и средств защиты, помогающих вам продвинуться вперед. Очень трудно назвать какое-то одно – по-нашему, лучше всех были два бонуса: огромный шар для боулинга, чтобы сшибать соперника, и вантуз, замедляющий движение конкурента перед вами. Трасса усеяна предательскими банановыми кожурками и взрывчаткой, наддающей скорости.

Одна из прелестей *STK* – включение в качестве гонщиков аватар из различных открытых проектов: например, динозавра с логотипа Mozilla и самого Тукса. Хотя эта версия – не столь уж значительное обновление. Новые карты и трассы выглядят неплохо, но настоящее изменение в том, что теперь *STK* поддерживает использование добавочных пакетов для дальнейшего расширения игры.

Чтобы игра работала на достойной скорости, потребуется хорошая видеокарта с самыми свежими драйверами. Кроме этого, перед компиляцией *STK* из исходных текстов надо будет установить *SDL* и *plib*.

В общем и целом, STK – очень приятная интерпретация игровой классики, и будет расти и развиваться в дальнейшем.

> «SuperTuxKart – очень приятная интерпретация игровой классики.»



› SuperTuxKart – отличная игра, с массой крутых персонажей и дьявольскими виражами.

Файловый менеджер

Gentoo

Версия 0.15.4 **Сайт** http://obsession.se/gentoo

До появления Gentoo Linux был файловый менеджер *Gentoo*. Основанный на древней двухпанельной парадигме, *Gentoo* вызывает у нас ностальгию, по многим причинам, но главная из них – очень точное воспроизведение основанного на Amiga файлового менеджера *Directory Opus*, вплоть до возможности назначать собственные кнопки быстрого доступа.

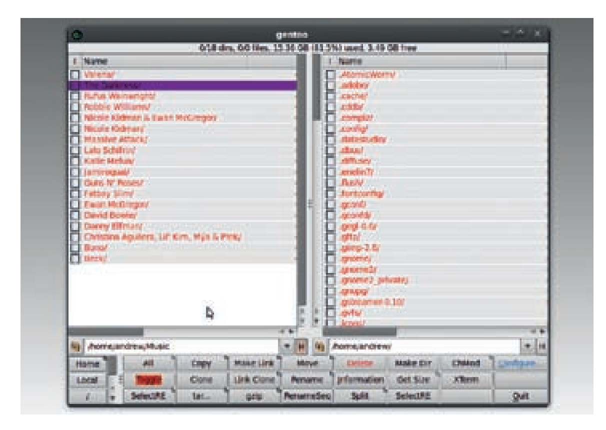
Предоставляется доступ к широкому диапазону полезных команд по умолчанию, а кому этого мало, при желании можно добавлять или удалять кнопки и даже задавать их цвет, чтобы они вписались в ваш стиль работы. С помощью интерфейса легко добавлять файлы в сжатые архивы, а также просматривать и извлекать их. А еще можно создавать ярлыки для наиболее часто используемых мест вашей файловой системы, чтобы улучшить навигацию.

Приложение сосредоточено на теме быстрого доступа: в строке заголовка даже приводится краткое описание, сообщающее о количестве директорий и файлов на вашей активной панели. Типам файлов приписывается определенный цвет, что помогает отличать их друг от друга, а также определенные действия. Для типизации файлов допускается использование регулярных выражений, и можно даже связать тип с определенным именем файла.

Чертовски быстр

К счастью, вся эта настройка отнюдь не тормозит *Gentoo*. Фактически, его красота – в его легковесности, и вам незачем

> «Для типизации файлов допускаются регулярные выражения.»



› Молниеносный и гибкий: да не Gentoo Linux, а впечатляющий файловый менеджер.

думать о зависимостях, поскольку он создан исключительно на C: просто запустите стандартные **./configure**, **make** и **make install**. Начав работать в *Gentoo*, вы, без сомнения, обнаружите, что перемещение по файлам и директориям оставляет далеко позади, например, *Dolphin* или *Nautilus*. *Gentoo* объединяет скорость терминала с преимуществами визуализации графического интерфейса. Добавить *Gentoo* к имеющимся у вас меню вам придется вручную, но, единожды им воспользовавшись, вы, возможно, обнаружите, что назад вам уже не хочется.

Астрономический инструмент

BoPlanets

Версия Сайт http://tinyurl.com/mj8glm

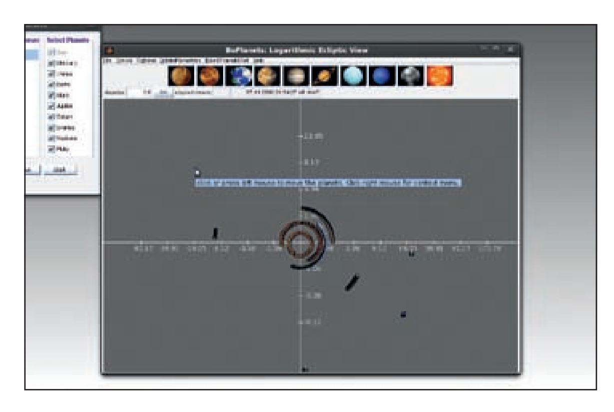
Мы здесь, в Башнях LXF, не только воспеваем свободное ПО; из нашего логова мы также восхищаемся величием небес. Поэтому мы с интересом встретили *BoPlanets* – небольшой опрятный инструмент для определения положения планет в Солнечной системе.

Прежде всего, знайте, что это – приложение Java, охотно работающее на любой поддерживаемой платформе. После запуска программа предложит выбрать, с какой планеты вы хотите вести наблюдение и какие небесные тела вы хотите отследить (нет, Сара Мишель Геллар [Sarah Michelle Gellar – актриса, сыгравшая главную роль в сериале «Баффи – истребительница вампиров», – *прим. пер.*] среди предлагаемых опций отсутствует). Нажав Start, вы попадете в главный интерфейс, где *BoPlanets* отображает точное расположение выбранных вами планет на текущее время и дату. С виду довольно скучно; но щелкните на центральной области и придержите кнопку мыши, и планеты начнут перемещаться вокруг вашего пункта наблюдения. По умолчанию дата изменяется с шагом в два дня вперед, но скорость изменения можно настроить с помощью поля Stepsize на панели инструментов. Мы предпочли отключить радиус-векторы, чтобы созерцать движение планет без лишних линий.

Взгляд на звезды

Орбиты планет можно видеть с трех разных точек, или же смотреть на них с Земли из любого выбранного вами города. Пути движения планет – на выбор – показываются в стандартном эллиптическом или логарифмическом виде, чтобы оценить относительное расстояние до каждого небесного тела. Потрясает выполняемый программой колоссальный объем работ по отображению объектов на карте: вы можете продвинуться вперед во времени на отрезок от кусочка дня до целого земного года и посмотреть, насколько сместятся планеты. Видеть небо будущего – просто уму непостижимо: когда вы узнаете, что очередное наложение орбит Нептуна и Плутона произойдет в середине XXIII века, это кажется фантастикой.

> «Объем работ по отображению объектов потрясает.»



› Это увлекательное приложение покажет небо далекого будущего, из любого города или даже с любой планеты.

В общем, BoPlanets – весьма интригующая программа. Проведите с ней пару минут, и вы потеряете счет времени.

Программа просмотра изображений

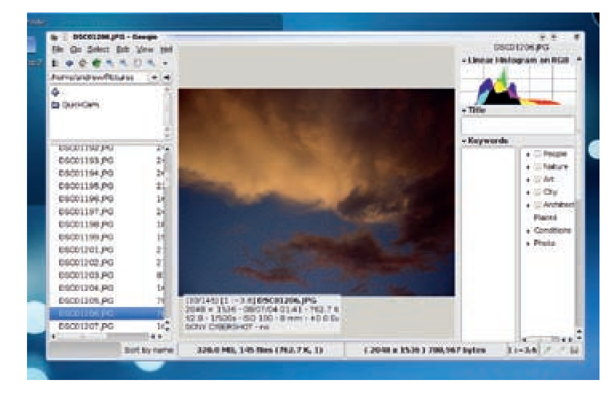
Geeqie

Версия 1.0 beta 1 **Сайт** http://geeqie.sourceforge.net

В наши дни стало нормой щелчком по файлу изображения получить либо его миниатюру, либо само изображение. Так, в Gnome для этой задачи используются *Gthumb* или *F-Spot*, а в других настольных средах в широком ассортименте представлены собственные небольшие и легковесные программки для просмотра изображений. И вам неважно, как это происходит – вы просто хотите видеть изображение, не озабочиваясь ничем другим.

С нашей точки зрения, *Geeqie* определенно попадает в категорию «беззаботных» программ – начиная со скромных зависимостей и заканчивая небольшим файловым браузером для беспроблемной навигации по большой библиотеке изображений. Львиную долю экрана занимает предпросмотр и информация, которая выводится под изображением. Информация Exif не предоставляется по умолчанию, но ее легко найти в меню View [Вид], и вы получите немалых размеров экран с подробнейшими сведениями о фотографии. Говоря об информации, *Geeqie* предоставит ее вам во множестве, а можно даже добавлять свои данные, снабдив каждое фото ключевыми словами или категориями. Это – симпатичное маленькое приложение, в частности, удобное для фотографов, и достойный соперник некоторым расхожим альтернативам.

› *Geeqie* — простая, легкая программа для просмотра изображений; она ничуть не путается под ногами.



Текстовый редактор

Minimum Profit

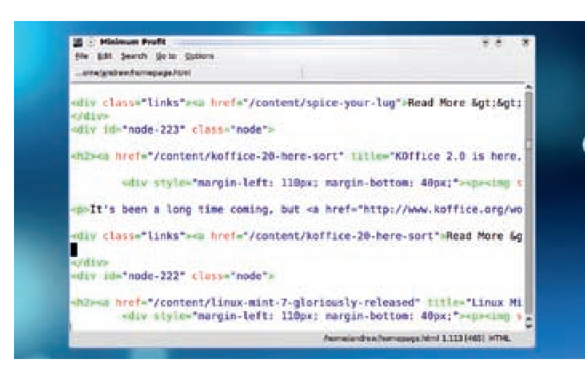
Версия 5.1.2 **Сайт** http://triptico.com/software/mp.html

Порой кажется, что программы просто из кожи вон лезут, заваливая нас бесчисленными панелями инструментов и пунктами меню. И все это изрядно грузит, поэтому мы горой стоим за простоту нашей рабочей среды.

Вот почему *Minimum Profit* (*MP*) вызывает вздох облегчения – это текстовый редактор специально для программистов и разработчиков, и вы не встретите здесь ни одной панели инструментов. Вместо этого все умещается в пяти пунктах меню, целиком оставляя центральную область восхитительно свободной для кодирования, кодирования и еще раз кодирования. Большинство пунктов меню вызывается с клавиатуры, так что, освоившись в *MP*, вы, возможно, вообще забудете о минималистском интерфейсе меню. Цветовое выделение кода основывается на расширении текущего файла, что помогает хорошей организации вашей работы, и каждый файл можно открыть в отдельной вкладке.

По части зависимостей, *Minimum Profit* многого не требует, и будет прекрасно чувствовать себя в любом из основных менеджеров окон. Более того, хотя интерфейс и прост, при желании копнуть поглубже вы найдете обширную документацию. Учитывая вышесказанное, мы охотно рекомендуем вам заменить *Vi* или *Emacs* на *MP*, и через пару месяцев вы о них и не вспомните. LXF

› *Minimum Profit* предоставляет чистое, незамусоренное пространство для кодирования.

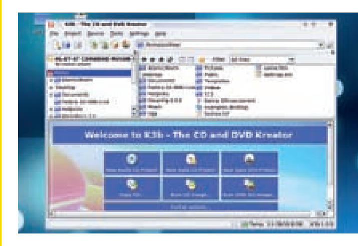


Также вышли

Новые и обновленные программы, тоже заслуживающие внимания...

K3b 1.66.0 alpha 2

K3b – приложение для записи CD/DVD, любимое фанатами KDE, с хорошей поддержкой записи CD и DVD и удобным интерфейсом пользователя. Но оно посложнее *Brasero*.
http://k3b.plainblack.com



› Фанаты KDE для записи дисков пользуются *K3b*.

Diffuse 0.3.3

Графический инструмент *diff* со ссылками на все достойные системы управления исходным кодом.
http://diffuse.sourceforge.net

Canorus 0.7

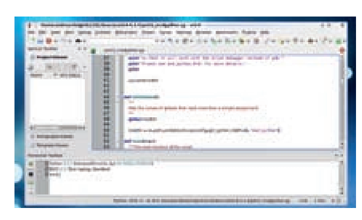
Canorus – редактор нотных записей по типу *Frescobaldi*. Он предлагает выбор фильтров импорта и экспорта и может использовать *Lilypond* для создания потрясающих партитур.
http://canorus.berlios.de

Eric 4.3.4

Эта почтенная IDE только что обзавелась обновлением, добавив массу опций к и без того уже внушительному списку функций.
http://eric-ide.python-projects.org

Jacl 2.4.8

Вернемся в былые дни текстовых приключений с помощью Jacl – языка, разработанного, чтобы помочь вам создать собственное приключение в стиле фэнтези.
http://freshmeat.net/projects/jacl



› Версия 4.3.4 *Eric* приобрела новые опции и функции навигации.

OBM 2.2.3

Ищете альтернативы для *Microsoft Exchange*? Тогда рассмотрите *OBM*, полнофункциональное приложение для групповой работы с хорошим web-интерфейсом.
http://obm.org

Начните здесь

Четыре потрясающих операционных системы плюс приложения и игры...



Читайте дальше — вы найдете всю необходимую информацию про DVD! Если вы новичок в Linux, откройте на диске файл index.html и перейдите в раздел Справка — там имеются руководства по ОС, в том числе:
- **» Что такое Linux?**
- **» Что такое дистрибутив?**
- **» Загрузка компьютера с DVD**
- **» Разбиение жесткого диска на разделы**
- **» Навигация по файловой системе**
- **» Учетные записи супер- и обычного пользователя**
- **» Работа в командной строке**
- **» Установка программ**
- **» Помощь онлайн**

...и еще много полезного!

Майк Сондерс
Редактор диска
mike.saunders@futurenet.com

Дистрибутив Linux

Linux Mint 7

Заняв почетное третье место в хит-параде популярности DistroWatch, Linux Mint стал одним из ведущих игроков в мире Linux. Он базируется на Ubuntu, но отличается иным подходом к интерфейсу, массой инструментов настройки и сообществом разработчиков, охотно принимающим в свои ряды. Мы спросили читателей www.tuxradar.com, за что они любят Mint – и вот некоторые из их ответов:

«Я полюбил этот новый дистрибутив. Он выглядит лучше, чем Ubuntu, и работает более ровно. Все меню более интуитивны, и дистрибутив выглядит профессиональнее многих» – Carlicus

«Очень радует открытость команды разработчиков Linux Mint для всех желающих участвовать в проекте. Я добавил функций в MintUpload, заново упаковал в пакеты, и мне в три дня разрешили выложить все это в репозиторий Community, при том, что это был мой самый первый опыт участия в FOSS. И эти улучшения теперь вошли в новый релиз!» – emorrp1

«Ubuntu меня несколько расстроил, а вот с Mint такого не произошло. Мне понравилось меню Mint, и на вид Mint намного приятнее» – mjjzf

Вы можете попробовать Mint с DVD этого месяца; он работает прямо с диска, но если вы захотите установить его на винчестер, следуйте подсказкам ниже. Требования Mint:

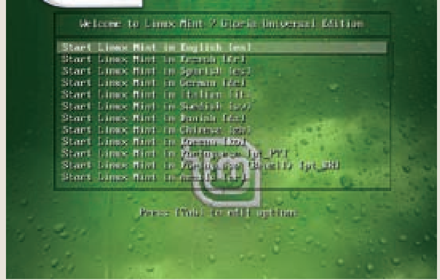
» **CPU** 1 ГГц x86

» **ОЗУ** 256 МБ (помните, что дешевые видеокарты отводят часть памяти под свои нужды!)

» **Жесткий диск** 10 ГБ свободного места

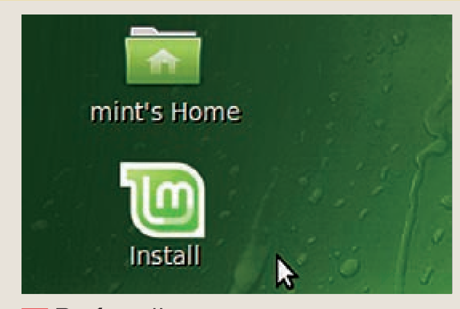
Читайте **user_guide.pdf** (на английском языке) в разделе **Дистрибутивы/Linux Mint** на DVD, чтобы получить более подробную информацию по Mint и добавочным утилитам, которые делают его уникальным. Версия на **LXF**DVD — 32-битная, но она с успехом будет работать и на 64-битных машинах. Также учтите, что это – Main Edition [Основная редакция]: в нее включены кодеки и проприетарные программы, так что вам не придется докачивать слишком много. Более детально – на www.linuxmint.com.

Шаг за шагом: Устанавливаем Linux Mint 7



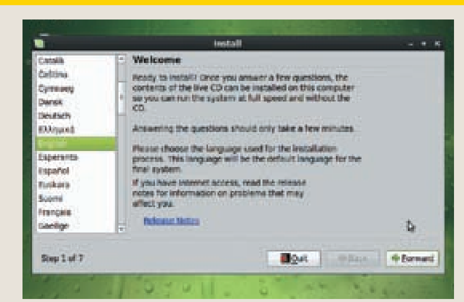
1 Загрузка

Запустите компьютер с DVD и нажмите на Enter при появлении меню (если потребуется изменить порядок загрузки, загляните в раздел **Справка/Новичку в Linux** на DVD).



2 Рабочий стол

Через несколько мгновений перед вами появится рабочий стол, где можно опробовать программы. После этого вы сможете дважды щелкнуть по значку Install [Установить], чтобы скопировать Linux Mint на жесткий диск.



3 Установка

Когда появится программа установки, выберите язык, раскладку клавиатуры и ваше местоположение. Если понадобится вернуться на шаг назад, вы всегда сможете нажать на кнопку Back.

Как бы мне?..

Установив Ubuntu с LXFDVD, вы, без сомнения, захотите узнать больше об использовании этого дистрибутива. Вот перечень наиболее общих задач и путей их решения…

» **Работа в сети** Нажмите кнопку Menu (внизу слева), и затем – Firefox, чтобы запустить самый популярный web-браузер с открытым кодом, поддерживаемый сотнями отличных расширений.

» **Чат онлайн** Войдите в Menu > All Applications > Internet > Pidgin. Эта программа поддерживает все популярные протоколы, в том числе AIM, ICQ, Yahoo, MSN и т. д.

» **Редактирование документов** Нажмите Menu > All Applications > Office, и затем выберите OpenOffice.org Word Processor (*Writer*) – текстовый редактор, Spreadsheet (*Calc*) – редактор электронных таблиц, или Presentation (*Impress*) – редактор презентаций; каждый из них совместим с документами *Microsoft Office*.

» **Редактирование изображений** Зайдите в Menu > All Applications > Graphics > Gimp – это самое мощное приложение для редактирования изображений в Linux.

» **Воспроизведение музыки и видео** Отправляйтесь в Menu > All Applications > Sound & Video и попробуйте *Rhythmbox* и *Movie Player*.

» **Добавление программ** Нажмите Menu и Software Manager, чтобы использовать супер-дружественный инструмент *MintInstall*, или используйте Package Manager, чтобы задействовать более продвинутую программу.

» **Работа в командной строке** Войти в нее можно через Menu > All Applications > Accessories > Terminal.

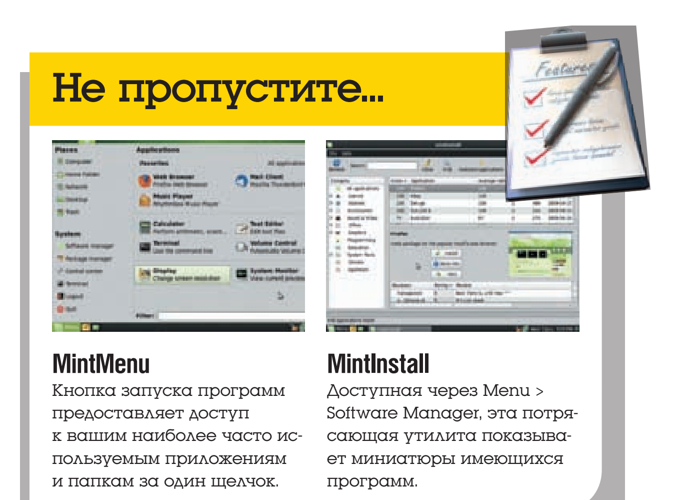
» **Настройка системы** Перейдите в Menu > All Applications, а затем – в Preferences или System, чтобы найти утилиты для смены интерфейса, настройки оборудования, управления пользователями и так далее.

Программа установки Linux Mint очень проста, но, с учетом миллионов комбинаций используемого компьютерного оборудования, всегда есть шанс возникновения проблем. Если рабочий стол не загружается, попробуйте удалить ненужную периферию (например, сканер или принтер), из-за распознавания которой могут возникнуть проблемы. Если вам понадобится помощь онлайн, на www.linuxmint.com вы найдете указания на документацию и списки рассылки. Можете также посетить Линукс-форум на www.linuxforum.ru и собственно форум Mint на www.linuxmint.com/forum. Удачи!



› **Цветовая гамма Linux Mint оттенка лакричника и лайма вызывает неудержимое желание лизнуть экран. М-м-м..**

Не пропустите...

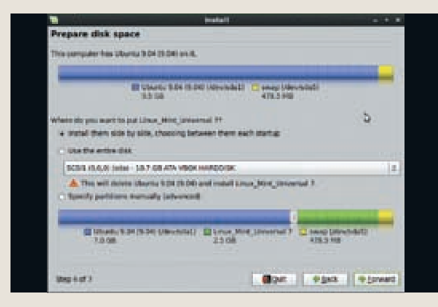


MintMenu

Кнопка запуска программ предоставляет доступ к вашим наиболее часто используемым приложениям и папкам за один щелчок.

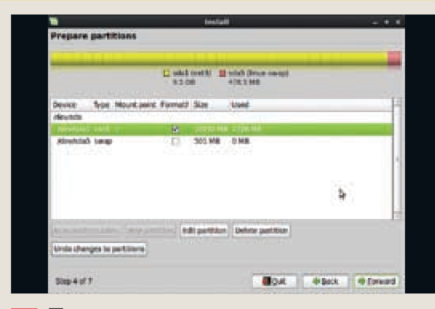
MintInstall

Доступная через Menu > Software Manager, эта потрясающая утилита показывает миниатюры имеющихся программ.



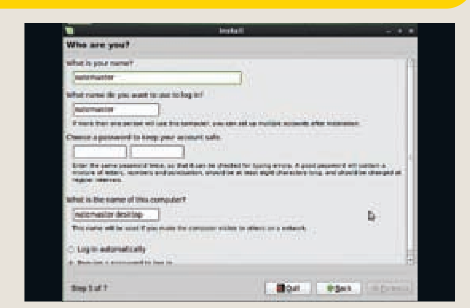
4 Деление диска на разделы

Программа установки предложит вам разбить ваш жесткий диск на разделы. Вы можете сжать имеющиеся разделы Windows/Linux, использовать диск полностью или разбить его на разделы вручную.



5 Деление диска на разделы вручную

Если вы решили разделить диск вручную, создайте корневой раздел (/) размером не менее 10 ГБ в формате ext3, и раздел подкачки (swap) размером 512 МБ для виртуальной памяти.



6 Создание учетной записи

Вы создадите пользовательскую учетную запись (в имени пользователя и пароле важен регистр); затем скопируются файлы Linux Mint, и можно будет перезагрузиться и зайти в новую систему.

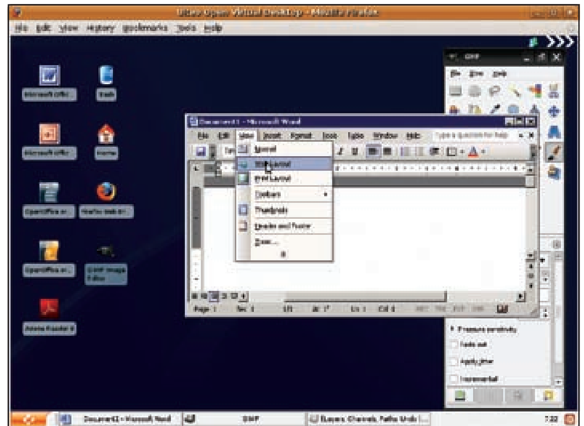
Другие программы

ОС, новые приложения, игры

Дистрибутив Linux

Ulteo OVD

Если вы еще не видели нашего обзора Ulteo, детища основателя Mandrake (ныне Mandriva) Гаэля Дюваля [Gaël Duval], загляните на с. 10 и сами увидите, о чем идет речь. Его ждали долго, но ждать стоило. На **LXF**DVD в разделе **Дистрибутивы/Ulteo** вы увидите **ovd-iso-latest.iso**, образ диска, который можно записать на DVD-R, а с DVD загрузить полную систему Ulteo с Менеджером сеансов и сервером приложений. Рекомендуем перед установкой прочитать **install.pdf**, чтобы получить общее представление о процессе, а в случае возникновения проблем обратитесь на www.ulteo.com/main/forums.



› Гаэль Дюваль долго трудился над Ulteo, и результат его трудов весьма впечатляет.

Дистрибутив Linux

EduMandriva 2009 Spring LXDE

Проект EduMandriva появился осенью 2007 года. За прошедшее время из простого репозитория он превратился в общероссийский образовательный проект, который охватывает несколько аспектов образовательного и технического направлений.

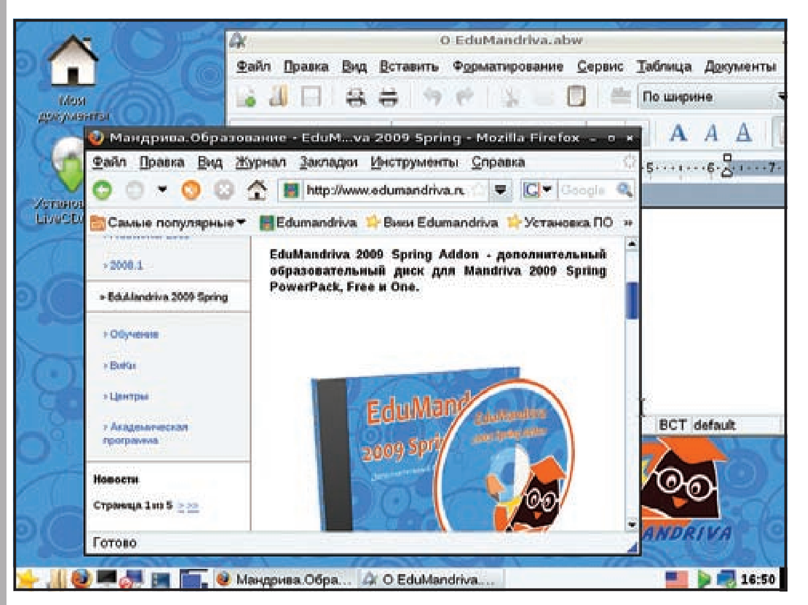
> «EduMandriva создают преподаватели для преподавателей.»

Отличительной особенностью EduMandriva является непосредственное участие в его разработке преподавателей, что позволяет говорить о создании образовательного ПО «преподавателями для преподавателей».

EduMandriva 2009 Spring LXDE представляет собой легковесный LiveCD на основе весеннего релиза Mandriva Linux (2009.1) и нетребовательной к ресурсам графической среды *LXDE*. Он предназначен как для работы без установки на жесткий диск, так и в качестве установочного CD/USB и основы для дополнительного образовательного диска.

Состав LiveCD позволяет использовать его как фундамент для знакомства с Linux и включает минимально необходимый набор образовательного ПО: текстовый процессор, электронную таблицу, растровый и векторный графический редакторы, среды и языки программирования Кумир, Pascal, Basic, среду для программирования и анимации Scratch, оболочку для тестов, клавиатурный тренажер и многое другое. Дистрибутив не требователен к памяти компьютера. Более подробную информацию о EduMandriva 2009 Spring LXDE можно получить в вики проекта по адресу: http://wiki.edumandriva.ru/index.php/EduMandriva_LXDE_ONE_2009.1_CD.

Чтобы попробовать дистрибутив в действии, возьмите ISO-образ из раздела **Дистрибутивы/EduMandriva 2009 Spring LXDE**, запишите его на чистую CD-R-мат-



› EduMandriva 2009 Spring LXDE можно использовать в режиме Live, а при желании — установить на жесткий диск.





Очень ВАЖНО!

Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим:

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу disks@linuxformat.ru для получения содействия.

рицу или вставьте в виртуальный привод *VirtualBox*. После этого перезагрузите компьютер. Чтобы установить систему на жесткий диск, щелкните по значку Установщик LiveCD/DVD, расположенному на рабочем столе, и следуйте инструкциям, появляющимся на экране. Обсудить EduMandriva можно на Линукс-форуме: www.linuxforum.ru.

Дистрибутив Linux

Russian Fedora Remix 11

Russian Fedora Remix 11 – это не новый дистрибутив, не ответвление и не клон. Это – респин Fedora 11, который команда Russian Fedora разработала специально для российских пользователей. Таким образом, устанавливая на своем ПК Russian Fedora Remix, вы получаете все преимущества Fedora 11: ускоренную загрузку, *Presto* (подключаемый модуль для *Yum*, позволяющий скачивать только изменения между пакетами и, таким образом, экономить трафик), переключение видеорежимов в ядре, файловую систему ext4 (по умолчанию), поддержку сканеров отпечатков пальцев, а также некоторые бонусы, призванные упростить использование дистрибутива на постсоветском пространстве.

Во-первых, в состав Russian Fedora Remix 11 включены мультмедиа-кодеки и проприетарные видеодрайверы, так что ваше оборудование, фильмы и музыка будут работать прямо из коробки. Во-вторых, возможность установки дополнительных пакетов с дистрибутивного DVD есть сразу же, даже в отсутствии сетевого соединения. Иными словами, в Russian Fedora Remix можно легко работать, даже если у вас нет быстрого и широкого канала в Интернет. Для тех, кто выходит в сеть через сотовый телефон, предусмотрены *NetworkManager* с *libmbca* (*Mobile Broadband Configuration Assistant*) и *Blueman*: по уверениям разработчиков, эта связка позволяет подключиться к Интернету одной мышью. Наконец, Russian Fedora Remix может не только создавать разделы ext4, но и загружаться с них.

Чтобы установить систему на жесткий диск, запустите компьютер со второй стороны **LXF**DVD (предварительно удостоверьтесь, что DVD-ROM стоит первым в списке загрузочных устройств) и следуйте экранным подсказкам. А потребуется помощь или захочется пообщаться – загляните на forum.russianfedora.ru.

› **Окно настройки *VirtualBox* позволяет контролировать память, используемую виртуальной системой.**

Настольные приложения

Paperbox и Me TV

Мы все больше и больше полагаемся на альтернативы традиционному файловому менеджеру для организации наших документов. Нам нужны настольные поисковые машины, способные осуществлять поиск внутри файла, чтобы отобразить его содержимое (вместо того, чтобы пытаться угадать имя), тэги и прочие крутые функции. *Paperbox*, созданный на движке Tracker, позволяет организовывать офисные документы, текстовые и прочие файлы с помощью тэгов и миниатюрных изображений и осуществлять навигацию в них. Особо впечатляющая функция – облако тэгов, в котором размер тэга зависит от частоты обращений к нему. Все это вы найдете в разделе **Рабочий стол**.

А *Me TV* – приложение для просмотра цифрового телевидения с DVB-устройств. Оно работает с картами DVB-t, DVB-C, DVB-S и ATSC и использует для вывода видеоплейер *Xine*. В качестве альтернативы, можно скомпилировать его для работы с *VLC*, *MPlayer* или *GStreamer*, чтобы он идеально соответствовал вашему любимому медиа-плейеру/движку. Помимо исходного кода, разработчики выпустили двоичный пакет для Ubuntu, он тоже есть на DVD.

Виртуальная машина

VirtualBox 3.0

Мы любим *VirtualBox*: это отличный эмулятор ПК, который позволяет устанавливать дистрибутивы Linux и другие ОС на виртуальный жесткий диск, и вам не приходится заново делить на разделы ваш настоящий винчестер. Вы можете использовать его, чтобы загрузить настоящие CD/DVD-диски или ISO-образы, так что если вы хотите испробовать Ulteo или еще какие-нибудь системы, не прожигая CD-R, установите его. Он находится в разделе **Система** на DVD: там есть пакеты для разных дистрибутивов, но вы можете просто выполнить run-файл для своей архитектуры от имени пользователя root.

Инструменты Интернет

BaShare и Twitim

BaShare проповедует замечательно прямолинейный подход к разделению доступа к файлам. Эта графическая программа создает HTTP (web) сервер на вашей локальной машине, а затем позволяет выбрать файлы, к которым вы хотите дать доступ своим друзьям по сети. Это очень полезно, если вы общаетесь с кем-то и вам надо быстро отправить ему файл, но ваша программа для чата этого не поддерживает (или файл слишком велик), и вам не хочется связываться с клиентом P2P. Лучше всего здесь настройка потребляемой ширины канала так, чтобы получатель вашего файл не забил бы его полностью.

Twitim – простой клиент Twitter, написанный на Perl, с *GTK*-интерфейсом. Среди его функций – настраиваемые списки просмотра, звуки, всплывающие уведомления и поддержка XMPP. Нам понравился ясный, ненавязчивый дизайн, и на него стоит обратить внимание, если вы заразились микроблоггингом.



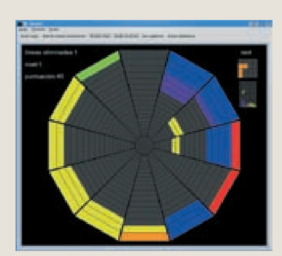
› **Twitim: все на Twitter, благодаря этому продуманному клиенту для микроблоггинга, на каком бы языке вы ни говорили!**

И напоследок...

Четыре отличных игры

Подумать только: исходная версия *Tetris* была написана 25 лет назад, но и по сей день остается одной из самых популярных игр планеты. *Quadra* – тщательно выполненная реализация *Tetris* со множеством захватывающих функций: игра через Интернет с несколькими соперниками, интегрированный CD-плейер, окно «удаленного просмотра» (можно подглядывать, что делают соперники) и многое другое. Командный *Tetris* – это всегда событие, так что агитируйте ваших онлайн-друзей тоже установить его, и начинайте играть.

Но! Если вам надоел обычный *Tetris*, попробуйте *Tubularix*, то есть (как вы, небось, уже догадались) «*Tetris* в цилиндрическом виде». Если вам трудно это представить, зайдите на сайт игры (http://tubularix.sf.net), там можно просмотреть видео. Если честно, от обычного *Tetris* он ушел недалеко – сверху сыплются блоки, и надо их уложить так, чтобы они совпадали. Но блоки можно перемещать также и за пределами колодца, и вам придется перестроить свой образ мышления, как если бы вы начали писать другой рукой.



› **Это *Tetris*, Джим, но не такой, как всегда.**

В разделе **Игры** предлагаем вам *Stendhal*, онлайн-ролевку, основанную на игровом движке *Arianne* и, судя по внешнему виду, вдохновленную ранней 2D-версией *Final Fantasy*. Кроме того, там имеется *MAX Reloaded*, реализация с открытым кодом классической игры 1996 года от Interplay под названием *Mechanized Assault and Exploration*. На DVD вы найдете пакеты для **Ubuntu** и **Fedora**, или можете скомпилировать исходный код, используя процедуру **./configure**, **make** и **make install** (от имени root). Подсказку по сборке программ вы найдете в разделе **Новичку в Linux** на **LXF**DVD. LXF

Информация о диске

Что-то потеряли?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, первым делом следует заглянуть именно туда.

Внимательно прочтите это перед тем, как использовать LXFDVD!

Форматы пакетов

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными двоичными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы могли собрать его самостоятельно.

Документация

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

Что это за файлы?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux и различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

» **имя_программы-1.0.1.i386.rpm** – вероятно, это двоичный пакет RPM, предназначенный для работы на системах x86;
» **имя_программы-1.0.1.i386.deb** – такой же пакет, но уже для Debian;
» **имя_программы-1.0.1.tar.gz** – обычно это исходный код;
» **имя_программы-1.0.1.tgz** – тот же файл, что и выше по списку: «tgz» – это сокращение от «tar.gz»;
» **имя_программы-1.0.1.tar.bz2** – тот же файл, но сжатый bzip2 вместо обычного gzip;
» **имя_программы-1.0.1.src.rpm** – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;
» **имя_программы-1.0.1.i386.FC4.RPM** – двоичный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;
» **имя_программы-1.0.1.ppc.Suse9.rpm** – двоичный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;
» **имя_программы-devel-1.0.1.i386.rpm** – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: disks@linuxformat.ru



Август 2009 LXFDVD 121 LINUX FORMAT

Август 2009 LXFDVD 121 LINUX FORMAT

DVD-приложение к журналу Linux Format № 8 (121). Август 2009

» Содержание

LINUX FORMAT

Сторона 1

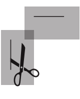
Сторона 2

Пожалуйста, перед использованием данного диска ознакомьтесь с опубликованной в журнале инструкцией!

КОММЕНТАРИИ? Присылайте ваши пожелания и предложения по электронной почте: info@linuxformat.ru

ДЕФЕКТНЫЕ ДИСКИ В маловероятном случае обнаружения дефектов на данном диске, обращайтесь пожалуйста по адресу disks@linuxformat.ru

Тираж изготовлен ООО «Уральский электронный завод». 620137, Россия, г. Екатеринбург, Студенческая ул., д. 9. Лицензия МПТР ВАФ 77-15.



Создание установочных дисков при помощи cdrecord

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Сначала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust! Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков: подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.



Пропустили номер?

» Мир свободного ПО богат и разнообразен, а потому далеко не все можно вместить в рамки одной статьи. Linux Format обходит эту проблему, публикуя серии статей по самым актуальным вопросам, но что делать, если вы поймали интересующий вас материал на середине? Обратитесь в Линуксцентр по адресу www.linuxcenter.ru и закажите желаемый номер журнала! Он доставляется как в печатной, так и в электронной форме, поэтому с момента открытия браузера и до получения нужного вам выпуска **LXF** может пройти не более нескольких минут!

Прямо сейчас для заказа доступны следующие номера:

LXF118
Май 2009

» Ищем идеальный дистрибутив Linux.
» *Cuneiform* и *Tesseract*: две ведущие открытые OCR-системы.
» Спутниковое ТВ: как настроить его в Linux и смотреть любимые передачи в *Kaffeine*, *VLC* и *MPlayer*.

LXFDVD: PCLinuxOS 2009.1, CentOS 5.3, *Clonezilla 1.2* и SystemRescueCd 1.1

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_118/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_118/

LXF119
Июнь 2009

» Будем продуктивнее: новый взгляд на знакомые приложения.
» Slack в будущее: в чем изюминка одного из старейших дистрибутивов Linux?
» Moblin: мобильная Linux-платформа от Intel с точки зрения пользователя и программиста.
» Java FX: новая технология от Sun Microsystems обещает легкий путь в мир богатых интернет-приложений.

LXFDVD: SimplyMEPIS 8.0, Slackware 12/Zenwalk 6.0, FreeBSD 7.2, OpenBSD/BSDanywhere 4.5 и NetBSD 5.0

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_119/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_119/

LXF120
Июль 2009

» Ubuntu стукнуло 10: взгляд в прошлое, настоящее и будущее.
» Intel и Linux: что такое Nehalem, VT-d, vPro и почему их поддержка появляется в Linux быстрее всех?
» *Bazaar*, *Subversion*, *Git*: в чем их сходства и различия, и какая система лучше.
» Каскад из CAD'ов: познакомьтесь с возможностями *OpenCASCADE* и *SALOME*.

LXFDVD: Ubuntu 9.04, Mandriva 2009.1 Free и подшивка Linux Format (номера 107–112)

Печатная версия:
http://www.linuxcenter.ru/shop/books-and-magazines/Linux-Format/lxf_120/

Электронная версия в формате PDF:
http://www.linuxcenter.ru/shop/electr/magazine/elxf_120/

Ну, а если вы хотите быть уверенными, что не пропустите ни один номер журнала – оформите подписку! Помните, что все подписавшиеся на печатную версию журнала через www.linuxcenter.ru получают электронную версию в подарок!

Спешите на www.linuxformat.ru/subscribe!





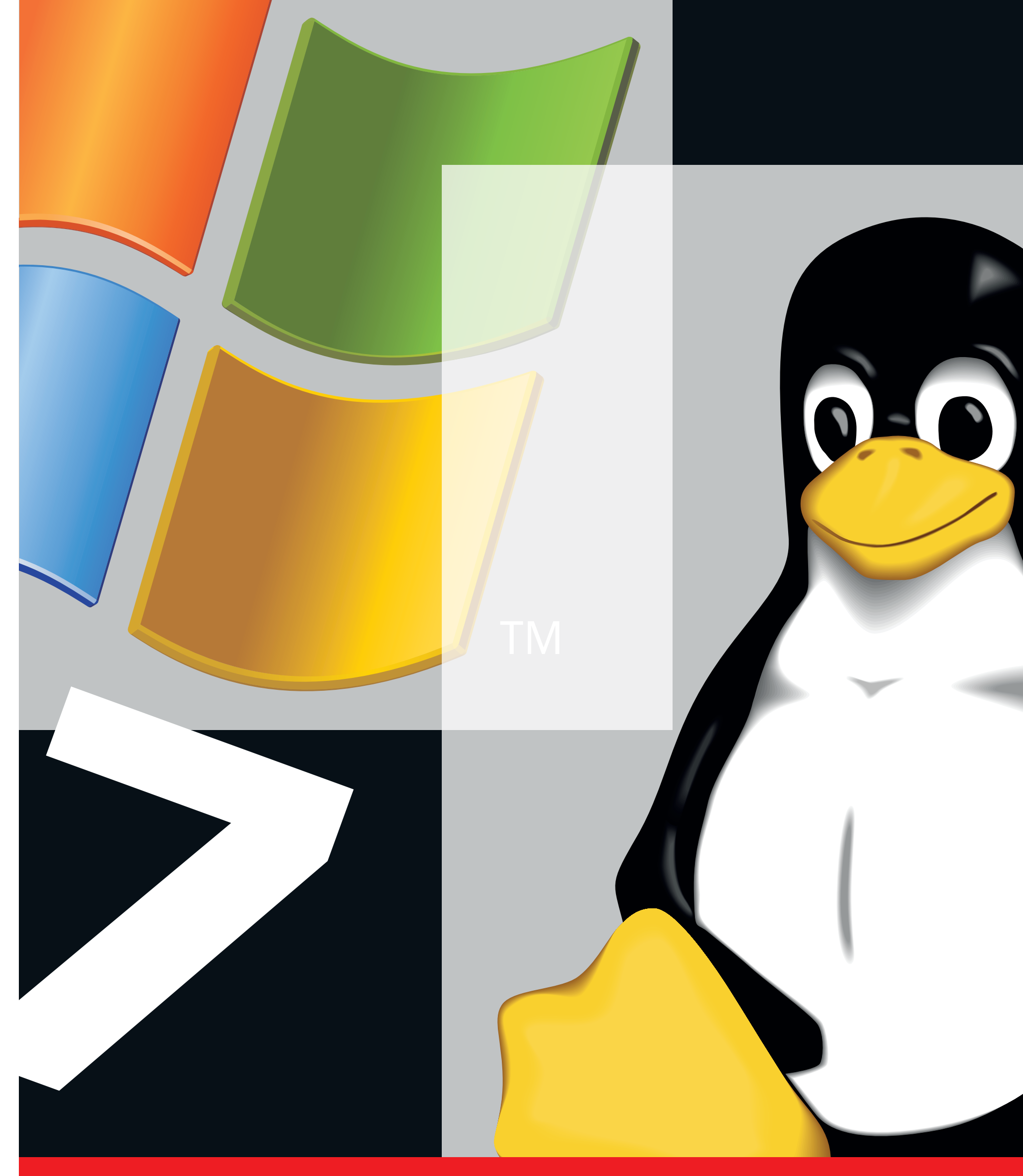
Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 5000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ
Главный редактор
Валентин Синицын info@linuxformat.ru
Литературный редактор
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Юлия Дронова, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развожаев, Татьяна Цыганова
Редактор диска
Александр Кузьменков
Верстка, допечатная подготовка
Сергей Рогожников
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Генеральный директор
Павел Фролов
Учредители
Частные лица
Издатели
Виктор Федосеев, Павел Фролов

Отпечатано в типографии ОАО «СПбГК»
198216, Санкт-Петербург, Ленинский пр., 139
Заказ ????

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ
Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Редактор обзоров Грэм Моррисон [Graham Morrison] graham.morrison@futurenet.co.uk
Редактор диска Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk
Художественный редактор Эфраин Эрнандес-Мендоса [Efrain Hernandez-Mendoza] efrain.hernandez-mendoza@futurenet.co.uk
Литературный редактор Эндрю Грегори [Andrew Gregory] agregory@futurenet.co.uk
Подготовка материалов
Нейл Ботвик [Neil Bothwick], Крис Браун [Chris Brown], Энди Ченнел [Andy Channelle], Марко Фиоретти [Marco Fioretti], Майкл Дж. Хэммел [Michael J. Hammel], Энди Хадсон [Andy Hudson], Мартин Мередит [Martin Meredith], Джульетта Кемп [Juliet Kemp], Саймон Пиксток [Simon Pickstock], Маянк Шарма [Mayank Sharma], Шашанк Шарма [Shashank Sharma], Ник Вейч [Nick Veitch], Евгений Балдин, Андрей Боровский, Евгений Зобнин, Артем Коротченко, Евгений Крестников, Николай Кузнецов, Дмитрий Смирнов, Алексей Федорчук, Игорь Штомпель
Художественные ассистенты: Стейси Блек [Stacey Black], Карлтон Хибберт [Carlton Hibbert]
Иллюстрации: Крис Хидли [Cris Hedley], Крис Винн [Chris Winn]

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel +44 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция):
Лиговский пр., 50, корп. 15
Тел. +7 (812) 309-06-86
Представительство в Москве:
Красноказарменная ул., 17, м. «Авиамоторная» (в помещении АТС МЭИ)
Тел/факс: +7 (499) 271-49-54
По вопросам сотрудничества, партнерства, оптовых закупок:
partner@linuxcenter.ru



В сентябрьском номере

Linux против Windows 7

Покажите своим друзьям, что новые функции в мире свободного ПО – это не обязательно покупка нового компьютера.

Анонимность в Сети

Быть инкогнито – значит, не давать никому отследить ваши перемещения по Всемирной паутине, как бы они ни пытались. Мы научим вас этому.

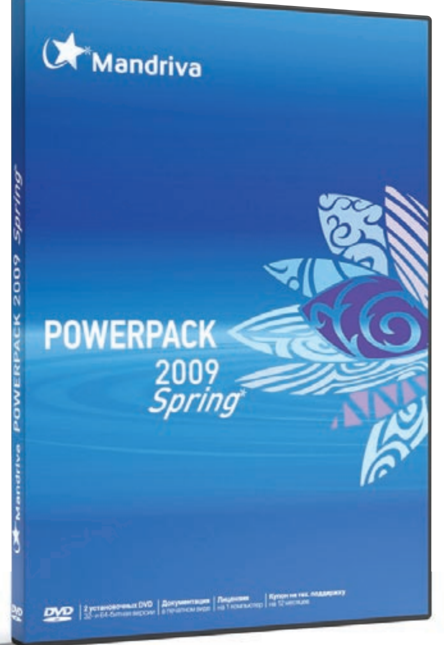
KOffice 2.0

После нескольких лет разработки *KOffice 2.0* наконец-то «ушел в золото» – но достаточно ли было сделано для победы над основным конкурентом, *OpenOffice.org*?

Содержание последующих выпусков может меняться без уведомления.



Mandriva Powerpack 2009.1 Spring



Дистрибутив Mandriva 2009.1 Powerpack включает в себя набор офисных и серверных приложений, и подходит для установки на офисной или домашней рабочей станции и на сервере. Mandriva 2009.1 Powerpack является отличным вариантом для миграции на GNU/Linux новых пользователей, и в то же время удовлетворяет запросы опытных пользователей и администраторов.

Совместимость с 1С

Входящий в дистрибутив wine@etersoft local дает возможность работы с популярными отечественными бизнес-приложениями (1С, «Гарант», «Консультант» и т. д.), также дистрибутив совместим с серверными версиями 1С для GNU/Linux.

Офисные приложения

В дистрибутив входит OpenOffice.org, интернет-приложения, графические, мультимедийные приложения, ПО для верстки и другие офисные приложения.

Служба каталогов

Кроме традиционных серверных приложений, в Mandriva Powerpack входит продукт Mandriva Directory Server (аналог контроллера домена Windows NT4) — простой в использовании инструмент для ведения каталога пользователей и прав доступа к общим ресурсам локальной сети предприятия и управления сетевыми сервисами (сервера DNS, DHCP, SAMBA, Proxy, Mail и др.).

Корпоративные продукты

Mandriva Directory Server

Mandriva Directory Server (MDS) — это простой в использовании инструмент для централизованного управления учетными записями и конфигурацией таких служб, как электронная почта, прокси сервер, служба доменных имен. MDS является основой для системы управления идентификацией пользователей, разграничения доступа пользователей к интернет ресурсам, квотирования почтовых ящиков, полностью заменяет Microsoft Windows NT4, IIS, m-Daemon, работает с Windows, Linux и Mac.

Linbox Rescue Server

Linbox Rescue Server (LRS) — пакет программ, предоставляющий функции локального и удаленного управления ИТ-инфраструктурой предприятия. LRS включает функции инвентаризации программного и аппаратного обеспечения, удаленного управления компьютерами и серверами, а также резервного копирования. Администратор может управлять ИТ-инфраструктурой через веб-браузер с любого локального или удаленного компьютера.

Сертифицировано ФСТЭК

Mandriva 2008 Spring Powerpack

Дистрибутив предназначен для рабочей станции или небольшого сервера, включает необходимые офисные, графические, мультимедийные, интернет-приложения и серверное ПО.

Mandriva Corporate Server 4 Update 3

Дистрибутив Mandriva Corporate Server предназначен для создания корпоративного сервера, на базе продукта можно создать интернет-сервер, веб-сервер, почтовый, сервер печати, сервер баз данных, сервер приложений и др.

Mandriva Flash

Mandriva Flash — дистрибутив GNU/Linux, загружающийся и работающий прямо с USB-носителя. Дистрибутив включает необходимый набор офисных приложений и допускает обновление и установку новых приложений, представляя собой полноценное мобильное рабочее место для работы на любом компьютере, поддерживающем загрузку с USB-устройства.

Наименование	Код Linuxcenter.Ru	Цена, руб.
Для рабочей станции		
Mandriva 2009.1 Spring PowerPack (DVD-box, техническая поддержка 12 месяцев)	lc4942	1900
Mandriva 2009.1 Spring PowerPack (DVD-box, техническая поддержка 6 месяцев)	lc4923	1300
Мобильное рабочее место		
Mandriva Flash 2008.1 (Mandriva Linux на 8 Гб USB накопителе)	lc3215	1875
Корпоративные продукты		
LRS Сервер (Linbox Rescue Server, платформы Linux/Windows) (лицензии на рабочие станции приобретаются отдельно)	lc2856	1690
Техническая поддержка на Mandriva Directory Server на 1 год	lc3560	5550
Лицензия и техническая поддержка на Mandriva Linux Powerpack 2009 на 50 компьютеров	lc3342	18 750
Пакет начальной поддержки Linbox Rescue Server	lc2821	99 000
Сертифицированные ФСТЭК продукты		
Сертифицированный ФСТЭК Mandriva 2008 Spring Powerpack	lc3408	2900
Сертифицированный ФСТЭК Mandriva Flash 2008.1 (на 4GB USB накопителе)	lc3409	4900
Сертифицированный ФСТЭК Mandriva Corporate Server 4.0 Update 3	lc3410	10 050

MANDRIVA LINUX ЯВЛЯЕТСЯ ОДНИМ ИЗ ПОПУЛЯРНЕЙШИХ МИРОВЫХ ДИСТРИБУТИВОВ ОПЕРАЦИОННОЙ СИСТЕМЫ GNU/LINUX

Главные преимущества дистрибутивов Mandriva — дружелюбный интерфейс, простота настройки, возможность легкого перехода пользователей с Windows на GNU/Linux, совместимость с широким спектром аппаратного обеспечения.

Дистрибутивы Mandriva Linux сертифицированы по требованиям ФСТЭК по 5 классу для СВТ и 4 уровню контроля НДВ, что дает возможность использовать их при работе с конфиденциальной информацией и персональными данными граждан.

Множество российских государственных и коммерческих организаций успешно используют Mandriva Linux на серверах и рабочих станциях. Среди них: Правительство Московской области, администрация Черниговского района Приморского края, Министерство финансов республики Саха (Якутия), группа компаний ИМАГ и многие другие.

С вопросами по продуктам Mandriva вы можете обращаться в Mandriva.Ru
Тел.: (812) 309-06-86, (499) 271-49-55
Электронная почта: info@mandriva.ru
www.mandriva.ru

Также в Mandriva.Ru доступны:

- литература по Mandriva Linux;
- услуги по установке, настройке и технической поддержке корпоративных продуктов Mandriva.